Date of acceptance Grade

Instructor

Content Monitoring in BitTorrent Systems

Tuomo Tilli

Helsinki September 12, 2012

Master's thesis

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Tuomo Tilli

Työn nimi — Arbetets titel — Title
Content Monitoring in BitTorrent Systems

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	September 12, 2012	70 pages + 2 appendices

Tiivistelmä — Referat — Abstract

BitTorrent is one of the most used file sharing protocols on the Internet today. Its efficiency is based on the fact that when users download a part of a file, they simultaneously upload other parts of the file to other users. This allows users to efficiently distribute large files to each other, without the need of a centralized server.

The most popular torrent site is the Pirate Bay with more than 5,700,000 registered users. The motivation for this research is to find information about the use of BitTorrent, especially on the Pirate Bay website. This will be helpful for system administrators and researchers. We collected data on all of the torrents uploaded to the Pirate Bay from 25th of December, 2010 to 28th of October, 2011. Using this data we found out that a small percentage of users are responsible for a large portion of the uploaded torrents. There are over 81,000 distinct users, but the top nine publishers have published more than 16% of the torrents. We examined the publishing behaviour of the top publishers. The top usernames were publishing so much content that it became obvious that there are groups of people behind the usernames. Most of the content published is video files with a 52% share. We found out that torrents are uploaded to the Pirate Bay website at a fast rate. About 92% of the consecutive uploads have happened within 100 seconds or less from each other. However, the publishing activity varies a lot. These deviations in the publishing activity may be caused by down time of the Pirate Bay website, fluctuations in the publishing activity of the top publishers, national holidays or weekdays. One would think that the publishing activity with so many independent users would be quite level, but surprisingly this is not the case. About 85% of the files of the torrents are less than 1.5 GB in size. We also discovered that torrents of popular feature films were uploaded to the Pirate Bay very fast after their release and the top publishers appear to be competing on who releases the torrents first. It seems like the impact of the top publishers is quite significant in the publishing of torrents.

ACM Computing Classification System (CCS):
C.2.2 [Network Protocols],
C.2.4 [Distributed Systems]

Avainsanat — Nyckelord — Keywords
BitTorrent, content, publishers, incentives, Pirate Bay, publishing rate

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

There is a vast amount of information being shared on the Internet today. Some people want to share their music, programs, pictures or videos with other people. Nowadays, most of the content is digitalized, so it is easy to make copies of it and share it with others. One problem in distributing this content to many people is that it requires a lot of resources like bandwidth. To address the issue of transferring files from one user to another, different kinds of technologies were developed.

The first player in peer-to-peer file transfer was Napster, released in 1999. Napster used central servers to index the users and find the files each user was holding. The central servers linked users to each other so they could transfer files among themselves. The central server was the main cause for the demise of Napster, when Napster was held liable for copyright infringement in 2001.

Napster is considered to be a first generation peer-to-peer technology. After Napster, came the second generation of peer-to-peer software, Gnutella, eDonkey2000 and Kazaa. They use a decentralized approach. Gnutella, released in 2000, was the first peer-to-peer system to use the decentralized model. When joining the network, a Gnutella servent connects to at least one node already in the network [12]. Now the servent can search the network by querying the nodes it is connected to. If a node has the file, it will report this to the querying node. At the same time, all of the queried nodes send the query to the nodes they are connected to. This way the query propagates to hundreds or thousands of nodes. All of the queries have a time to live (TTL) field, so the whole network is not flooded with the query messages. Gnutella is still used today.

eDonkey2000 software was released in September of 2000. Contrary to the Gnutella protocol the eDonkey2000 protocol uses servers [13]. The eDonkey network uses a client/server model. The servers are run by power users and do not share any files. They just hold the information about the shared files and their respective client locations. A client joining the network connects to a server via TCP. The client sends information about itself and the files it offers for distribution to the server. The server provides a list of other known servers to the client. Now the client is connected to the eDonkey network and can search and download files. Files are downloaded directly from other users. eDonkey was officially shut down in 2005 following a cease and desist letter from the Recording Industry Association of America (RIAA). However, the eDonkey network is still in use by applications like

eMule.

In Kazaa, introduced in 2001, the users connect to each other directly without the use of a central server [21]. Some of the nodes act as super nodes and the rest are ordinary nodes. Each ordinary node is connected to a super node and super nodes are connected to a number of other super nodes. Super nodes have information on all of the files each ordinary node connected to it is holding. This allows super nodes to assist in finding the files searched within the network. Kazaa is no longer in use.

In 2001, the BitTorrent protocol, a very efficient technology to get content distributed to others was released. BitTorrent is one of the most used file sharing protocols on the Internet today [14]. Its efficiency is based on the fact that when users download a part of a file, they simultaneously upload other parts of the file to other users. This allows users to efficiently distribute large files to each other, without the need of a centralized server.

Sharing content using BitTorrent is easy. The distribution of files in BitTorrent is handled with the help of .torrent files. When a user wants to get a file, he first obtains the relevant .torrent file. The .torrent files include information on the pieces of the file and the address of the tracker that manages the swarm. When the user connects to the tracker, it receives information on the number of seeders and leechers and a number of random IP addresses of other participating peers in the swarm. The user can use these IP addresses to connect to the swarm and start exchanging information with the other peers.

BitTorrent sites are very popular. The most popular torrent site is the Pirate Bay with more than 5,700,000 registered users [33][27]. These BitTorrent sites do not contain any of the shared material. They only contain information on how to contact other users who are sharing the data. This information is usually in a .torrent file. However, for example, the Pirate Bay does not even store .torrent files anymore [27]. The discovery of distributed hash tables (DHTs) enables the use of Magnet links. Magnet links are just hashes, strings of characters, so users do not actually have to download anything from the torrent site.

The BitTorrent swarms create ecosystems that can contain hundreds or even millions of peers. The content distributed using the BitTorrent protocol include, among others, music, porn, movies, games and software. Some of this material is copyrighted and in many countries illegal to distribute. This raises the question why people are uploading so much copyrighted content when they could face legal consequences.

In this research, we will investigate the BitTorrent ecosystem. We will go over the previous research on the content distributed using BitTorrent. We will also look into the initial publishers of the content and their publishing habits and examine the incentives of the publishers.

The motivation for this research is to find information about the use of BitTorrent, especially on the Pirate Bay website, which will be helpful for system administrators and researchers. We will focus our research on the content published, the publishing patterns, the file size distribution and the behaviour of the top publishers of torrents. This should provide us with an overview on the load the Pirate Bay website is on from the uploads of torrents and allow us to develop the system further.

We have data on all of the torrents uploaded to the Pirate Bay from 25th of December, 2010 to 28th of October, 2011. We use this data to inspect the kind of content distributed with BitTorrent using the Pirate Bay website. We look into the behaviour of the biggest publishers found from the data. We also use the data to check how fast torrents of new movies show up on the Pirate Bay compared with the release dates of the movies.

What is interesting is that only a very small portion of the users publish most of the content. This is very clear from the results gained from previous researches as well as the results obtained from the Pirate Bay data. According to the studies [8], the users who have published the most content, are advertising their own sites on the description pages of the torrents they have uploaded. This seems to be profitable, because they are doing it although they could face criminal charges.

We also investigated the publishing speed of ten movies released in 2011. Fake torrents of the selected movies showed up on the Pirate Bay, in some cases, even months before the premiere of the movie. Also, good quality versions of the movies appeared on the Pirate Bay web site even before the DVD of the movies had been released in most regions.

This paper is organized as follows. Chapter two gives information on Peer-to-Peer networking and BitTorrent. We also go over briefly the eMule application. The chapter discusses the BitTorrent protocol in detail and some enhancements made to BitTorrent. We will also go over what BitTorrent is used for, what kind of content is distributed and who are the major publishers of the content and what are their incentives.

Chapter three will discuss how the data was collected from the Pirate Bay web site

and what kind of database was used in this research.

In chapter four we present the results gained from the Pirate Bay data on the torrents uploaded to the Pirate Bay. We will go over what kind of content is being distributed. We take into closer investigation the top publishers, who have uploaded the most torrents to the Pirate Bay during the time frame that the data covers. We investigate how the sizes of the files are distributed among the torrents, the publish activity of torrents and the top publishers' publish patterns. Lastly, we present results on the rate at which torrents are uploaded to the Pirate Bay website.

Chapter five will go over the results on the investigation of the ten selected feature films. We will compare the dates of the torrents uploaded to the Pirate Bay with the release dates of the movies. We will also take a look at who are the publishers of the torrents for the ten selected movies.

Chapter six will conclude the thesis.

2 BitTorrent And Its Use

First this section defines what Peer-to-Peer networking is. In section 2.2 we will introduce a Peer-to-Peer application named eMule. Then in section 2.3 we will describe the BitTorrent protocol in detail. Section 2.4 describes enhancements made to BitTorrent. Lastly, in section 2.5 we will show what BitTorrent is used for and what kind of research has been done on the subject. We will describe the methods used to find the initial publishers of content, what are their incentives and what kind of content is shared using BitTorrent.

2.1 Peer-to-Peer Networking

Peer-to-Peer networking differs from the traditional Client/Server networking. In Client/Server networking one entity acts as the provider of resources or services. This entity is the server. Then there are one or several other entities who request content or services from the server. These are the clients. The definition of Client/Server networking can be seen in Definition 2.1.

Definition 2.1 [31] Client/Server network is a distributed network in which there are entities, who provide resources or services and entities, who request

the resources or services from the servers. The resource providers are the servers and the resource requestors are the clients. The servers are the only providers of resources. The clients do not share any of their resources.

In Peer-to-Peer networking the participants all share a portion of their resources with each other. All the participants, the peers, act as resource providers as well as resource requestors. The peers can access each other directly. The main difference between Client/Server networking and Peer-to-Peer networking is that the clients in Client/Server networking do not share any of their resources. Peer-to-Peer networking is defined in Definition 2.2.

Definition 2.2 [31] A distributed network is a Peer-to-Peer network if the participants share their resources with each other. The participants, the peers, act as resource requestors and resource providers. The peers can contact each other directly without intermediary middlemen.

Peer-to-Peer networking can be divided into two categories. Peer-to-Peer networks with central entities and networks where all the entities are equal. Peer-to-Peer networks with central entities are called Hybrid Peer-to-Peer networks and networks without central entities are called Pure Peer-to-Peer networks. These two networks are defined in Definition 2.3 and 2.4.

Definition 2.3 [31] A distributed network is a Hybrid Peer-to-Peer network, if it is a Peer-to-Peer network according to Definition 2.2 and it requires to have at least one central entity to provide the services offered by the network.

Definition 2.4 [31] A distributed network is a Pure Peer-to-Peer network, if it is a Peer-to-Peer network according to Definition 2.2 and any single node can be removed from the network without the network suffering from loss of service.

2.2 The eMule Application

The eMule application was released in 2002 as an alternative for the eDonkey2000 client. The eMule application is based on the eDonkey2000 client, released in 2000.

eMule is still updated and widely in use.

Each file in the eMule network is given a file ID [10]. The file IDs are used in identifying the files in the network and also in the detection of corrupted files. The file ID is a hash value calculated from the content of the file. First the file is divided into parts, 9,28 MB each. Then the MD4 algorithm [25] is used to calculate the hash value of each part. The file ID is calculated by combining the hashes of each part and calculating the MD4 hash of the result. If the file is less than 9,28 MB, then the file ID is the hash of the only chunk.

An eMule client maintains an upload queue for each of its files being shared. When a user requests a file from another client, the user will be put into the bottom of the queue. Once the user reaches the top of the queue, it can start downloading the file, one part at a time. A client can download different parts of the file from different users. Similarly to the BitTorrent protocol, once a client has downloaded one part completely, it can upload it to other users requesting it.

eMule implements a credit system. The credit system is used to reward users who upload content to the network. The more credit a user has, the faster it will advance in their waiting queues.

In the credit system, each user is given a user ID. The ID is 128 bits long and it is created by concatenating random numbers [18]. However the value of the 6th byte is 14 and the value of 15th byte is 111. The user ID is unique and is used to identify users across different sessions.

eMule offers two different networks. Firstly, there is the classic server based eD2k network and secondly, a serverless network based on Kademlia [10]. Both of the networks offer the same services of searching for files and finding the sources of the files.

2.2.1 The eD2k network

The eD2k network is composed of eMule servers and eMule clients [18]. When joining the network, an eMule client first connects to an eMule server using TCP. The eMule client is preconfigured with a list of eMule servers. An eMule server has a database containing information on all the clients connected to it and the files they are sharing.

Upon connecting to a server the client receives a client ID from the server. The ID is valid through the client-server connection's lifetime.

The client IDs are divided into low and high IDs. The client receives a low ID if it cannot accept incoming connections, which may be the result of the client's firewall blocking the connections. This restricts the client's use of the eMule network and some servers might reject the client's connection. The low ID given to clients differs from server to server and is always lower than 16777216 [18].

An eMule client is provided with a high ID, if the client allows other clients to freely connect to its TCP port. An eMule client with a high ID may use the eMule network with no restrictions. High IDs are calculated in the following way [18]:

Lets assume the client's IP address is A.B.C.D. Then the ID will be

$A + 2^8 * B + 2^{16} * C + 2^{24} * D$

using the big endian representation.

A user can search the eD2k network for keywords in filenames in two ways [10]. First the search can be local. This means the search uses information only from the server the user is connected to. Secondly the search can be global, in which case the search will use information from all of the servers connected to the network. A local search is faster than a global search, but a global search will likely provide more results than a local one.

When a server receives a keyword to be searched, it will look up the keyword in its database. If there are matches, the server will provide the searching client with a list of file names, the hash value of the file and the clients it knows for having the file that contain the keyword. Now the user has the information needed to connect to a client holding the searched file. All users that have at least one entire part of the file are considered to be sources.

2.2.2 The Kademlia network

A user wanting to connect to the Kademlia serverless network will need the IP address and the port number of a client already connected to the network. Once connected to the client, the user request other clients to determine if it can accept incoming connections. This is the same as the server checking if the client can accept incoming connections. If the user can accept incoming connections, it will be given an ID and an open status. Otherwise, if the user cannot accept incoming connections, it will be given a firewalled status.

Searching and finding sources to download from in the Kademlia network and the

Kademlia protocol is explained thoroughly in section 2.4.2.

2.3 The BitTorrent Protocol

BitTorrent is a very widely used peer-to-peer file distribution protocol. The protocol can be considered to be a Hybrid Peer-to-Peer network, if it uses a centralized tracker. On the other hand, if a decentralized tracker is used, the network can be considered to be a Pure Peer-to-Peer network. BitTorrent is estimated to account for 40%-70% of Internet traffic, depending on the region [14]. BitTorrent's success can partly be due to its openness. The BitTorrent protocol has been publicly published [4], so anyone can create their own client applications.

Rank	Client
1	uTorrent
2	Azureus
3	Mainline
4	BitComet

Table 1: Four popular BitTorrent clients.

When a user wants to distribute a file or a group of files using the BitTorrent protocol, the user first needs to create a .torrent file. There are many BitTorrent clients that can be used to create .torrent files. Four popular BitTorrent clients can be seen in table 1 [34]. The .torrent files contain information that is needed to get the actual data. The data in the .torrent files include the URL of the tracker, hashes of the pieces of the file, to verify the integrity of the pieces and additional information like names of the files and the piece length used [4]. The structure of a .torrent file can be seen in table 2.

The .torrent file can be distributed through, for instance, email or a BitTorrent portal, to anyone who wants to download the file(s) specified by the .torrent file. A list of ten of the most popular BitTorrent portals can be seen in table 3 [33]. Alexa [1] and Compete [6] are websites that rank websites according to their popularity.

The user wanting to distribute a file makes the file available by acting as a seed. Anyone who has the complete file can act as a seed. Others downloading the file are called peers or leechers.

A user who wants to share a file needs to do the following [4]:

Key	Description
announce	The URL of the tracker.
info	This maps to a dictionary with keys described below.
name	This is the suggested name to save the file as.
piece length	The number of bytes in a piece.
pieces	This is a string whose length is a multiple of 20. The string is divided into strings of length 20. Each string is a SHA-1 hash of the piece at the corresponding index.
length	The number of bytes of the file being shared. This is only present if there is only one file being shared.
files	This is only present if there are multiple files being shared. It maps to a list of dictionaries containing the keys described below.
length	The number of bytes in the file.
path	A list of strings corresponding to subdirectory names, the last of which is the actual name of the file.

Table 2: Torrent file structure.

1. Start a BitTorrent tracker (or use an existing one).
2. Create a .torrent file using the files to be shared and the URL of the tracker.
3. Provide access to the .torrent file.
4. Start BitTorrent and act as a seed by sharing the files.

A BitTorrent tracker is a server that provides information about the users downloading content using the BitTorrent protocol. Users contact the tracker to obtain contact information for peers downloading the contents of the corresponding torrent. Peers also contact the tracker from time to time get the contact information of new peers and also to provide statistics.

In order to know what parts of the files each downloader has, the original file(s) is divided into smaller pieces [5]. For example a 10 MB file could be divided into twenty 512 kB pieces. Peers wanting to download the file(s) specified by the .torrent file connect to the seeds or peers and start downloading pieces of the file. The peers connect to each other using TCP. All the downloaders report to each other what pieces they have. The users verify the integrity of the pieces by checking that the

Site	Alexa Rank	Compete Rank
The Pirate Bay	78	577
Torrentz	142	1 053
KickassTorrents	257	1 165
IsoHunt	259	1 153
BTjunkie	385	802
ExtraTorrent	446	1 574
Demonoid	455	2 397
EZTV	1 151	13 647
Bitsnoop	1 415	5 242
1337x	2 006	10 856

Table 3: Ten of the most popular BitTorrent sites.

SHA-1 hashes of the pieces match the hashes in the .torrent file. Once the integrity of the piece is checked, the peer can start uploading the piece to other peers. This way the piece is not only available from the seed, but also from other peers who have downloaded the piece. This enables the file to be distributed among peers very efficiently. This technique of dividing the files into smaller pieces allows users to pause and resume downloads easily.

A peer can find the other peers sharing a file by obtaining the IP addresses of the peers. This happens by contacting the tracker(s). The URL of the tracker(s) can be found from the .torrent file. When the client contacts the tracker, the tracker returns a number of random IP addresses of peers or seeds distributing pieces of the file(s). Every once in a while the client contacts the tracker again to get the IP addresses of new peers who might provide faster download times. This also allows new clients to connect to the swarm.

Figure 1 demonstrates the steps for a client to download a file using BitTorrent.

1. A user surfs the web to find a file it wants.
2. The user downloads the relevant .torrent file.
3. User opens the .torrent file with a BitTorrent client. The client contacts the tracker over HTTP and sends information on the file it is downloading and the port number it is listening on.

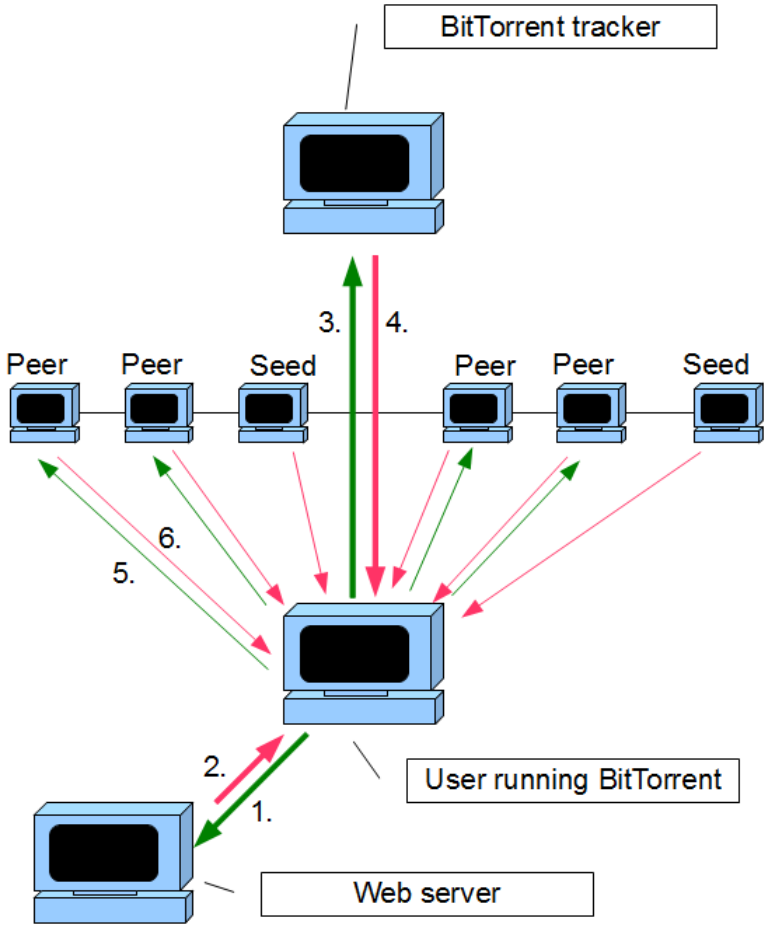

Figure 1: Phases taken by a user to obtain a file using BitTorrent.

4. The tracker responds with a list of contact information for the peers who are downloading the same file.

5. The downloader contacts other peers and lets them know which pieces of the file it wants. The piece selection is explained in more detail in the next section.

6. The peers respond by sending the piece of the file requested.

The BitTorrent protocol offers no privacy to its users. All the users connected to a swarm can be identified by obtaining their IP addresses from the relevant tracker. This enables, for instance, authorities to identify users who are uploading copyrighted material.

2.3.1 Piece Selection

The order in which the peers select a piece to download is very important for good performance [5]. Using a poor algorithm for piece selection could result in all the peers having the same pieces and in the worst case the rarest pieces could become unavailable. BitTorrent solves this issue by using the rarest first piece selection algorithm [5].

In the rarest first piece selection algorithm the peers select the rarest piece available to download next. The peers know all the pieces the other peers, they are connected to, have, so they can select the piece which the fewest of the peers have. This strategy lowers the risk of some pieces becoming unavailable due to peers or seeds departing the swarm.

Only exception to this technique is in the beginning of the download. This is when a peer has no pieces. At this point it is important for the peer to get any piece as fast as possible. Rare pieces are probably slow to get so in the beginning a peer selects a piece to download at random. Once the peer obtains the first complete piece, it switches to the rarest first strategy.

The rarest-first algorithm does a good job of making sure that each peer has a piece that the other peers want [20]. This in turn offers more possibilities for the peer selection algorithm.

2.3.2 Peer Selection

In BitTorrent, each peer is responsible for maximizing its own download rate [5]. The peers do this by downloading from the peers they get the fastest download rates from. Each peer reports to the peers they are connected whether they are interested or not and choked or not. Being interested means that the other peer has a piece the user wants. Being choked means that the user will not upload to that peer. The transfer of data happens only when one peer is interested and the other peer is not choking. However, a peer can download from a choked peer, as long as the choked peer has them unchoked. In the beginning of the connection the peers are choked and not interested [4], so the task of the peer selection algorithm becomes which peers to unchoke.

The decision which peers to unchoke are purely based on download speeds [5]. The download speeds are calculated using the average speeds over the last 20 seconds.

The BitTorrent peers check who to choke every ten seconds. This is enough for TCP to get to the full transfer speeds. Once every ten seconds a fixed number (default is four) of the fastest peers, who are interested, are unchoked.

Only unchoking the peers by their provided download rates offers no way to know if other peers would provide even faster download rates. This is why every 30 seconds a peer uses the optimistic unchoke. In optimistic unchoke a peer unchokes an interested peer at random. The optimistic unchoke allows the peer to become aware of new faster connections and also it allows new peers, who have just joined the swarm, to get their first piece.

Once a peer has downloaded all the pieces, it becomes a seed. In seed state the peer can no longer use the download rates of other peers for peer selection. In the original peer selection algorithm the seed preferred the peers to whom it had the fastest upload speeds to. However, in the newer versions of BitTorrent the algorithm was changed [20]. The new algorithm works as follows:

1. All of the unchoked and interested peers are ordered according to the time they were last unchocked, so that the most recently unchoked peer is first in line. This is done every ten seconds.

2. For the next two 10 second intervals the three first unchoked and interested peers in the list are unchocked and an additional fourth interested and choked peer is unchoked at random.

3. The first four peers are kept unchoked for the third period of ten seconds.

This new algorithm ensures that the peer selection in seed state stays fair [20].

2.4 Enhancements to BitTorrent

In it's original form BitTorrent networks can be considered to be Hybrid Peer-to-Peer networks. They require the use of a central tracker in order to facilitate the finding of peers connected to the swarm. This makes the network vulnerable to faults. The central tracker could encounter denial of service attacks or even legal actions could be taken against the administrator of the tracker. If the tracker goes down for some reason, the users can no longer obtain the IP addresses of the other peers connected to the swarm. Also, the use of a centralized tracker generates traffic to the site hosting the tracker which might generate costs for the administrator. The

solution to these problems is to use a decentralized tracker and by using magnet links one does not even have to download .torrent files to be able to obtain the wanted content. Decentralized trackers are made possible with the help of distributed hash tables (DHTs). Distributed hash tables and magnet links are explained in the next sections.

2.4.1 Magnet Links

Magnet links enable users to find and download content using BitTorrent without the need of .torrent files [11]. The users are able to join a swarm and download the metadata from other peers connected to the swarm. Metadata refers to the info-dictionary part of .torrent files. The info-dictionary part of .torrent files includes the suggested name to save the file as, the length of the pieces, the SHA-1 hashes of the pieces, the length of the file and the subdirectory names, if there are any.

The format of magnet links is the following [11]:

magnet:?xt=urn:btih:<info-hash>&dn=<name>&tr=<tracker-url>

The parameter **"xt"** means **"exact topic"** and it is the only mandatory parameter in the magnet link. It is a uniform resource name (URN) [21] formed from the cryptographic hash value of a file.

<info-hash> is the cryptographic hash of the file in question, for a total of 40 characters. For compatibility issues there is also support for the 32 character base32 [2] encoded hash.

The parameters **"dn"** and **"tr"** are optional. **"dn"** refers to **"display name"**, which is the name displayed to the user. **"tr"** is the tracker url. Multiple trackers can be identified with multiple "tr" values.

If there is no tracker url specified, the client can use DHT to find the peers.

2.4.2 Distributed Hash Tables (DHTs) and Kademlia

A distributed hash table is a distributed data structure that performs the function of a hash table. In a hash table a hash function is used to map keys to their associated values [7]. Storing an element with key k in a hash table works as follows:

1. A hash function h is used to calculate the hash h(k).
2. The element key is stored in slot h(k) of the hash table.

Distributed hash tables work in a same way, except the hash table is distributed among different nodes in the network.

Distributed hash tables are used in BitTorrent for storing peer contact information [9]. This allows peers to find each other without the use of a central tracker. The function of the tracker is distributed among the peers.

When this decentralized tracker strategy is used in BitTorrent, each user runs two protocols in parallel. First, they act as peers running the BitTorrent protocol to transfer files. Second, they act as nodes implementing the distributed hash table protocol to get the location of peers.

The distributed hash table protocol used in BitTorrent uses the UDP protocol and is based on Kademlia [9]. In Kademlia [24], each node is assigned an ID. The node ID is a 160-bit long string of characters chosen at random from the same 160-bit space as BitTorrent infohashes, for example a SHA-1 hash of some data. The ID is created when the node joins the network.

When a node joins the network, certain keys assigned to the node's successor become assigned to the node. Nodes' closeness to each other is calculated using the XOR (exclusive or) metric interpreted as a positive integer. The XOR metric calculates the distance of two nodes by taking the bitwise exclusive or of the two nodes' IDs.

Each node stores keys that are close to the node's ID, meaning that the XOR value of the hash of the key and the node's ID is small. When a node wants to find peers, it compares the infohash of the torrent with the IDs of the nodes in its routing table. Then it contacts the nodes which are closest to the infohash and requests the contact information of the peers downloading the torrent. If the contacted node does not know any peers downloading the torrent, it returns the contact information of the nodes in its routing table that are closest to the infohash of the torrent. Then the original node queries the new closest nodes and this goes on iteratively until it cannot find any closer nodes or the contact information for peers downloading the content is found.

Each return value for a query for peers includes an opaque value known as the "token". In BitTorrent the token is created using the SHA-1 hash of the IP address concatenated to a secret that changes every five minutes. When a node announces that it is controlling a peer that is downloading a torrent, the node must present the token received from the same queried node in a recent query for peers. The queried node checks the token against the querying node's IP address. The token is used to

prevent malicious hosts from signing up other hosts for torrents. Tokens up to ten minutes old are accepted.

The routing table of each node contains the contact information for a number of good nodes. The contact information includes the triple <IP address, UDP port number, Node ID> for each node. A node is "good", if it has responded to one of our queries within the last 15 minutes or it has responded to one of our queries and has queried us within the last 15 minutes. If a node has been inactive for 15 minutes, it becomes questionable. If the node does not answer to multiple queries in a row, it becomes bad.

The entire node ID space from 0 to 2^{160} is covered by the routing table. The routing table is divided into parts called k-buckets. Each k-bucket is a list containing at most k nodes (k = 8 in BitTorrent by default). Each list contains nodes that are a certain distance from the node.

At first the routing table is empty and there is only one k-bucket with an ID covering the entire node ID space range from 0 to 2^{160}. When a node with ID N is inserted into the routing table, it is put into the k-bucket that covers the ID space range of N. Once a k-bucket becomes full of good nodes, no more nodes can be added to it. If the k-bucket covers the range of our own node ID, the k-bucket is split into two new k-buckets and the nodes are divided among them. If the k-bucket does not cover our own node ID, and it is full of good nodes, the new node is discarded. If a k-bucket is full and it contains bad nodes, one of the bad nodes is replaced with the new node. If there are questionable nodes in the k-bucket, the least recently seen node is pinged. If that node responds, then the next least recently node is pinged until one fails to respond and that node is replaced or all the nodes in the k-bucket are known to be good and no node is replaced. This ensures that the routing table is filled with good long running nodes.

The probability of a node remaining online increases with the amount of time the node has been online [30]. Since Kademlia tries to keep the oldest nodes in the k-buckets, it maximizes the probability that the k-buckets contain nodes that will remain online. This method also provides protection against attacks where the adversary flushes the nodes' route tables by flooding the network with new nodes.

The nodes in Kademlia are treated as leaves in a binary tree. Each node's position is determined by the shortest unique prefix of its ID. This is shown in figure 2. Figure 2 shows the three k-buckets for our selected node 011. Nodes 111, 110, 101 and 100 are in the farthest k-bucket. If k was 3, only three of these nodes would be in the

k-bucket and known to node 011. Nodes 000 and 001 are in the second k-bucket and node 010 is in the third k-bucket. Each node has more knowledge of nodes that are close to it than of nodes that are far from it, since there can be more k-buckets in the routing table for the nodes close to the node.

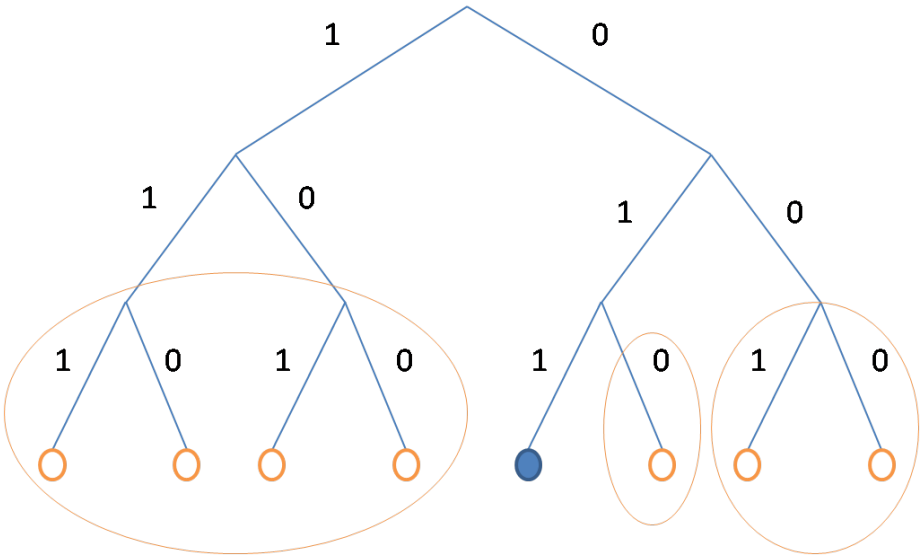

Figure 2: An example tree of nodes in a network.

The Kademlia protocol uses four remote procedure calls (RPCs). The RPCs are the following:

1. **PING** - checks if a node is online.
2. **STORE** - instructs a node to store a <key, value> pair.
3. **FIND_NODE** - instructs a node to return the <IP address, UDP port number, Node ID> triplet for k nodes it knows about closest to the 160-bit ID given as an argument. If the node does not have knowledge of k nodes, it returns the triplets for the nodes it knows about.
4. **FIND_VALUE** - behaves like FIND_NODE, except that if the node receiving the RPC has received a store RPC for the value, it returns the stored value.

Configuration information is transferred alongside the RPCs. The k-buckets are often kept fresh by the traffic of requests travelling through the nodes.

The most important procedure for a node is to find the k closest nodes to some ID [24]. This procedure is called a node lookup. When initiating a node lookup, the node picks α nodes closest to the ID from its own k-buckets. Then the node sends FIND_NODE RPCs to the α nodes it has selected. α is a system-wide parameter, for instance 3.

Once the initiating node receives information on new nodes from the previous RPCs, it resends the FIND_NODE RPC to the new nodes. This is done recursively and new RPCs can be sent before all α of the previous RPCs have returned. The initiating node picks α nodes from the k nodes closest to the target it has not yet queried and resends the FIND_NODE RPC to them. If a node does not respond, it is removed from consideration. If a round of FIND_NODE RPCs does not return any nodes closer than already has been seen, the initiating node sends the FIND_NODE RPC to all of the k closest nodes it has not queried yet. The lookup procedure terminates when the initiating node has queried and received reponses from the k closest nodes it has seen.

The node lookup procedure is used by most of the operations. When a node wants to store a <key, value> pair, it uses the node lookup procedure to locate the k closest peers and sends them the STORE RPC.

When a node wants to find a <key, value> pair, it starts the node lookup procedure. However, instead of using the FIND_NODE RPC the node uses the FIND_VALUE RPC and the procedure stops as soon as the wanted value is returned.

When a node wants to join a network, it must have the IP address and port number of one node in the network. This node is called a bootstrap node. The joining node inserts the bootstrap node into the appropriate k-bucket and performs the node lookup procedure on its own node ID. This allows the joining node to obtain information on other nodes and populate its k-buckets. Also, other nodes will become aware of the new node.

2.5 The Use of BitTorrent

The BitTorrent protocol was designed for the efficient distribution of large files. Instead of the traditional Client/Server model, BitTorrent uses the Peer-to-Peer model in which the bandwidth stress is divided among the users. This is the cornerstone of the popularity and adoption of BitTorrent.

There are studies which investigate the users of BitTorrent and what kind of content

is distributed using BitTorrent [8][34][19]. Also the performance of the BitTorrent algorithm has been studied [20][3][16][28]. In the next section we will provide the results from the previous studies on what kind of content is distributed using BitTorrent and who are the content providers.

2.5.1 Research Methods

The process of publishing content via BitTorrent can be divided into two parts.

1. First, a user wanting to share a file creates a .torrent file. Then the .torrent file is published via, for instance, a torrent portal for others to download.
2. Second, the user must contact the tracker and let it know that it has the complete original file. Now the user acts as a seed and the file(s) is available. Now when other peers get the .torrent file and contact the tracker they will be informed of the seed.

This information can be used to identify the initial publisher. Some of the torrent sites offer RSS feeds to announce new torrents or they have a dedicated page for new torrents. Once we notice that a new torrent has been published, we can quickly connect to the tracker and see if there is only one peer distributing content. This first peer connected to the swarm distributing the content must be the initial publisher.

Sometimes there are more than one peer subscribed to the tracker. The content could have been distributed first among a private community. In this case one has to use other information to identify the initial publisher of the content. One strategy is to use the login information of the content providers.

On some torrent sites, like the Pirate Bay, users are required to login before they can upload content. One can use this login information to identify the torrents that a user has uploaded. For instance, Le Blond et Al. [19], use the method of sorting the content by username. Then they check if one IP address is uploading content in many of the torrents uploaded by that username. If the IP address is found, then that is identified as the initial publisher of the content. However, this method did not prove to be very efficient in identifying the initial publishers.

Cuevas et al. [8] use the RSS feed to detect a new torrent. Once a new torrent is detected they immediately download the .torrent file and subscribe to the associated tracker. If there are less than 20 users connected to the swarm and there is only one

seed, they identify the seed as the initial publisher. However, in many cases there were many peers connected to the swarm already or the initial seeder was behind a NAT box, so the IP address of the initial publisher could not be identified. Using this technique Cuevas et al. were able to obtain the publishers' usernames for all of the torrents and the IP addresses of more than 40% of the initial publishers.

The number of peers in a swarm and the IP addresses of them can be obtained by querying the tracker. When a tracker receives an announce started request, the tracker returns a subset of the peers and the number of seeds and leechers distributing that content. Thus after querying the tracker enough times one can get the IP addresses of each peer connected to the swarm. This allows an adversary to identify all the seeds and leechers connected to the swarm.

2.5.2 Major Publishers

In their research [8], Cuevas et al. study what kinds of users publish content in BitTorrent and why. They monitor two big BitTorrent portals, the Pirate Bay and Mininova. They get data of more than 55 000 different published content and over 35 million IP addresses.

To identify the publisher of a new torrent, Cuevas et al. [8] use the method of following the RSS feeds offered by the monitored BitTorrent portals. The RSS feed has information like the content category, the content size and the publisher's username of the .torrent file. They leverage the RSS feed to detect a new .torrent file. To get the publisher's IP address they immediately download the .torrent file and connect to the tracker. If there is only one seeder in the swarm, it is the initial publisher of the file. Using this method, they were able to identify the publisher's username for all of the torrents and the publisher's IP address in 40% of the torrents. To obtain the IP addresses of the other participants in the associated swarm, they periodically query the tracker.

Cuevas et al. [8] use primarily the username to identify individual publishers since the username is expected to remain the same across different torrents. They observe that the top 3% of the BitTorrent publishers contribute about 40% of the published content, as can be seen in figure 3. Also, about 40% of the top-100 publishers do not download any content at all and 80% download less than 5 files. They also observe that a substantial number of major publishers are located at a few hosting services.

These results seem similar to the results obtained by Le Blond et Al. [19]. They

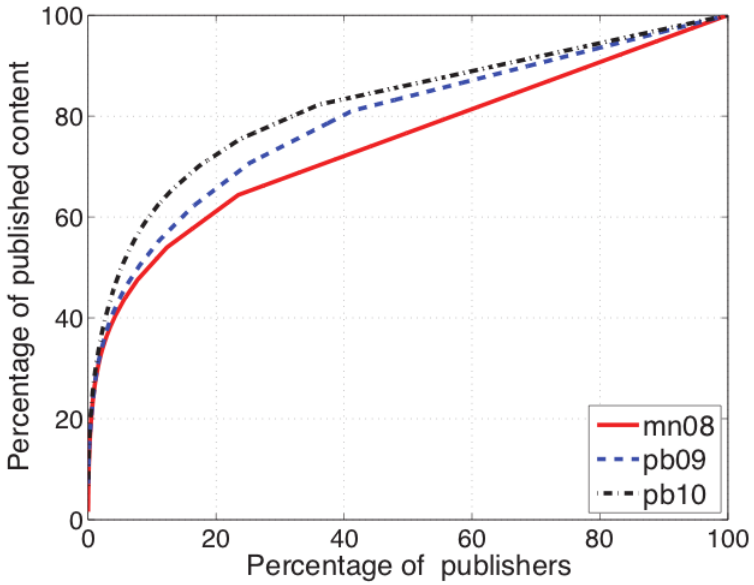

Figure 3: [8] Graph shows the percentage of published content by a percentage of publishers. mn08 is data gathered from Mininova in year 2008, pb09 and pb10 are data from the Pirate Bay in years 2009 and 2010.

concluded that the biggest content providers inject 30% of all the contents and the top 1 000 content providers inject 60% of all the contents injected into BitTorrent.

Focusing on the top-100 content publishers, only 55% of them are used by a unique username. The remaining 45% of IP addresses of the content publishers map to a large number of usernames. After careful examination of this set of IP addresses, Cuevas et al. [8] discovered that they either use manually created or hacked accounts to inject fake content. By fake content they mean torrents of which the content does not match the description of the torrent. These fake torrents might contain anti-piracy messages or malicious software.

The uploaders of fake content seem to be associated with anti-piracy agencies or are malicious users, who distribute malware. These fake publishers are responsible for about 25% of the usernames and 30% of the published content. The fake publishers primarily focus on video and software content. The anti-piracy agencies publish fake versions of recent movies and the malicious users publish software that contains malware.

The top-100 content publishers, excluding the publishers of fake content, can be divided into three groups, as can be seen in figure 4. The first group, 25% of the top publishers, have their own BitTorrent portals that in some cases use private

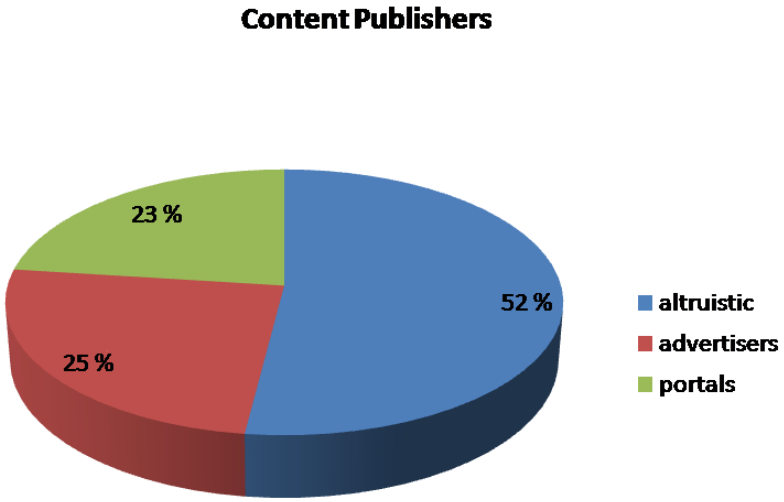

Figure 4: Top-100 content publishers divided into three groups.

trackers. These publishers provide 18% of the content and are responsible for 29% of the downloads.

Second group, 23% of the top publishers, advertise some URLs of web sites. They publish 8% of the content and are responsible for 11% of the downloads.

The remaining group, 52% of top publishers, seem to be altruistic users. These users do not advertise any URLs. They are responsible for 11.5% of the content and 11.5% of the downloads.

In conclusion, a small number of publishers are responsible for 67% of the published material and 75% of the downloads [8]. These publishers are either fake publishers who publish fake content or belong to the top publishers of which about half advertise a web site in their published torrents and the other half seem to be altruistic.

2.5.3 Content Distributed Using BitTorrent

BitTorrent is used in the distribution of all kinds of data. To make it easier for users to browse torrent portals the torrents are usually put into one of four categories:

video, audio, software or others. Video mainly includes movies, tv-shows and porn content. Category others includes, among others, e-books, comics and pictures. Cuevas et al. [8] studied how the amount of content distributed is divided among different categories for all of the torrents, for the fake torrents and for the torrents published by the top publishers, excluding the fake content. Figure 5 shows the percentages of torrents in each category for all of the torrents. As one can see, most of the files are video files and then audio, software and others contribute almost an equal share.

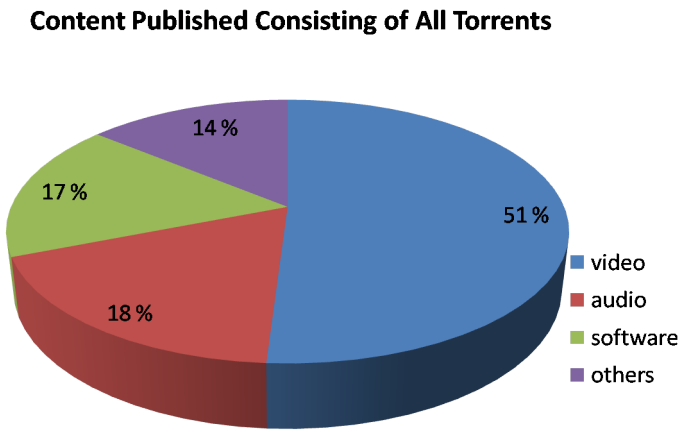

Figure 5: Percentage of torrents in each category for all of the published torrents for the data set collected by Cuevas et al. [8].

Zhang et al. [34] studied also the content being distributed using BitTorrent. They use data from five major torrent-discovery sites, Mininova, The Pirate Bay, BT-monster, Torrent Reactor and Torrent Portal. They found 1 192 203 active torrents. Figure 6 shows the amount of torrents in each category. Zhang et al. [34] have divided the content into more categories than Cuevas et al. [8]. However, if movies, tv-shows and video is added we get 39% for video content. This is a bit less than the 51% from the other study. If music and audio is added together for the audio category, we get 25% which is a bit more than the 18% from the other study. For the software category we add application and games categories and get 16% which is very close to the 17%. This leaves the category others which contributes 20% of all of the torrents.

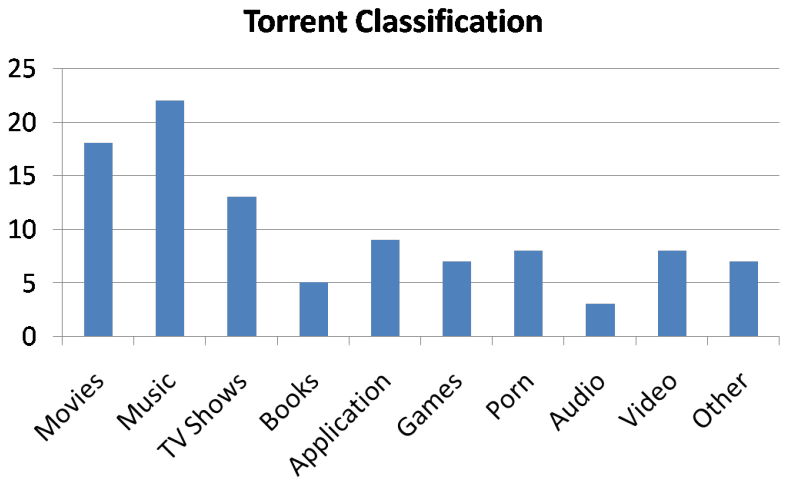

Figure 6: Percentage of torrents in each category for all of the published torrents for the data set collected by Zhang et al. [34].

Figure 7 shows the percentages of each category for the fake content. The fake content seems to be focused on video and software content. This makes sense since the anti-piracy agencies publish fake versions of recent movies and tv-shows and malicious users publish software that contains malware.

Figure 8 shows the percentages for all torrents published by the top publishers, excluding the fake content. Most of this content is video files and also audio takes a big share. On the other hand there is not much software content being published by the top publishers. This could be explained by people recording tv-shows and movies on their television. Apparently people are also sharing a lot of music.

Mateus et al. [23] studied the amount of content transferred using BitTorrent for each category. They calculated the number of copies of content transferred for each content type. The results can be seen in figure 9. As can be seen in the figure, the amount of transferred copies is a bit different from the amount of content published using BitTorrent.

All in all, it seems like most of the content being shared by all of the publishers is video files. One reason for the popularity of video files might be that, for instance, some tv-shows may not be available for viewing in different areas so people have no other way of watching them than by downloading them. Also new movies could be available for download even before they are showing in your area. Video content is also easy to provide just by recording it on you television. The next chapter will take a look at the incentives of the top publishers.

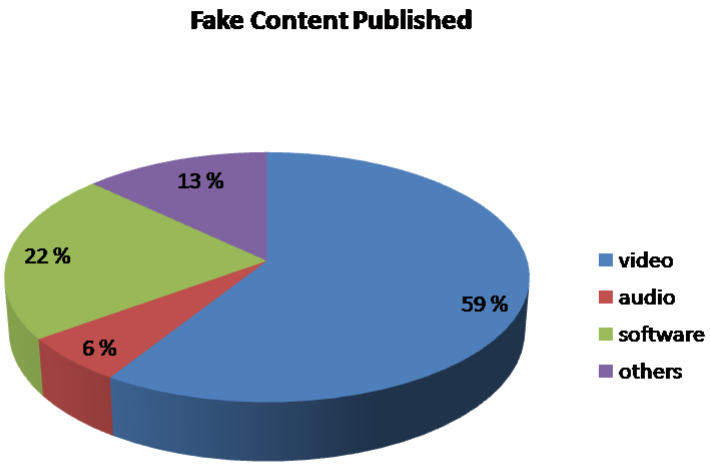

Figure 7: Percentage of torrents in each category for the fake content.

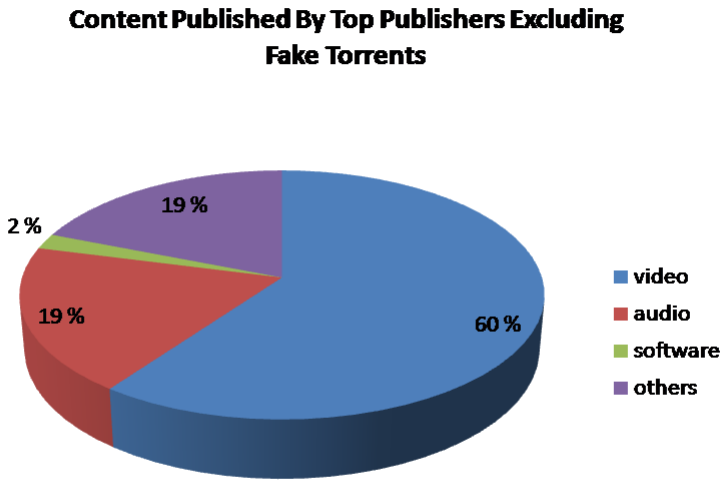

Figure 8: Percentage of torrents in each category for the top publishers, excluding the fake content.

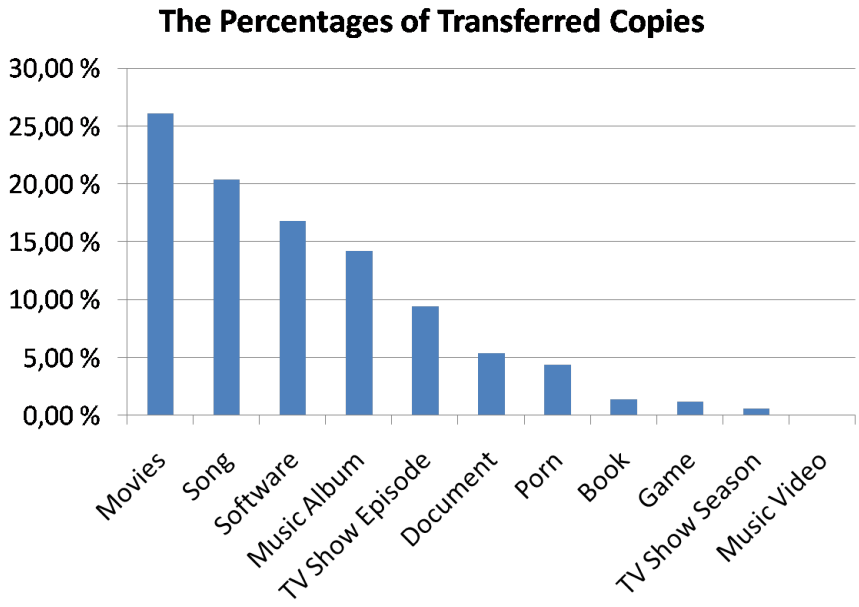

Figure 9: Percentages of copies transferred using BitTorrent by type of media.

2.5.4 Incentives of the Content Publishers

A very small number of users (∼100) produce 67% of the content, which is 75% of the downloaded material [8]. Most of these users only upload content and rarely download content published by others. The published content is often copyrighted, so the publishers could very well face legal actions toward them. This brings up the question: what are the incentives for publishers of the content?

To find out the answer to this question, Cuevas et al. [8] divide the major publishers into two groups, fake publishers and top publishers. Fake publishers publish a large number of fake content and top publishers publish a large number of often copyrighted material. The incentives of the top publishers are explored and they are also divided into three categories: (1) Private BitTorrent Portals, who produce income through ads, donations or fees, (2) Publishers who promote a website to attract users to visit there and (3) Altruistic major publishers. These categories of users are characterized and the values of the web sites associated with each publisher are estimated to get a view of the incentives.

Cuevas et al. [8] examine the incentives of the two groups of publishers, the fake publishers and the top publishers. The fake publishers provide content with catchy titles, but the contents of the files are fake. Some of them include an anti-piracy message and others point to malware software. So the incentives of the fake pub-

lishers are to distract the distribution of copyrighted content and in some cases to lead users to download malware.

The other group of publishers, the top publishers, publish non-fake and often copyrighted material. They allocate a lot of resources to do so. This group of top publishers can be divided into three groups. One group, 25% of the top publishers, own their BitTorrent portals that in some cases use private trackers. The private trackers offer better download rates to users, but require the user to upload a certain amount also. These publishers gain financial profit in three ways: (1) through advertisements on their web sites, (2) through donations from visitors and (3) through collecting a fee for VIP access.

Second group, 23% of the top publishers, advertise some URLs of web sites. Most of these users (70%) publish porn content and they often advertise an image hosting web site. These portals get their income from advertisements.

The remaining group, 52% of top publishers, seem to be altruistic users. These users do not advertise any URLs.

All in all, about half of the top publishers seem to be advertising a web portal in their published torrents. They try to attract a large number of users to their web sites and create income through ads, fees and donations. The profit-driven publishers provide 26% of the content and 40% of the downloads. Other research [34] on the topic has been done and the results of those studies seem to support most of these results.

3 The Pirate Bay Data Set

We collected data on all of the uploaded torrents from the Pirate Bay web site for the period of 25th December 2010 to 28th October 2011. The exact data collected is described in section 3.1 and the database used in handling of the data is described in section 3.2.

3.1 Data Collection

We collected data on all the new torrents uploaded to the Pirate Bay. A crawler was written in Python to fetch the data.

We use two nodes for the sampling, both Dual Intel Xeon E5440 @ 2.83GHz with quad cores, 32 GB memory and Gigabit connection to the Internet. The operating system is Debian SMP with Linux 2.6 kernel. On each node, we set up a crawler with its own sampling policy. One monitors publish activities, the other monitors the evolution of users (number of seeds, leechers, and swarms). The sampling frequency is twice per minute.

The Pirate Bay has a web page which lists all the new torrents uploaded to the Pirate Bay. At the time of writing this, the URL of the page for recent torrents was http://thepiratebay.se/recent. If the crawler noticed that a new torrent had been uploaded, it would fetch the data related to that torrent.

Figure 10 shows a screenshot of one of the torrent description pages the crawler would get the data from. The spots where the data was taken from are circled in the figure. The fetched data includes the following items:

1. **infohash** - The 20-byte SHA1 hash of the info dictionary part of the .torrent file.
2. **type** - The type of the content.
3. **subtype** - The subtype of the content.
4. **infourl** - The URL of the page shown in figure 10.
5. **name** - The name of the torrent.
6. **torrenturl** - The URL of the torrent file.

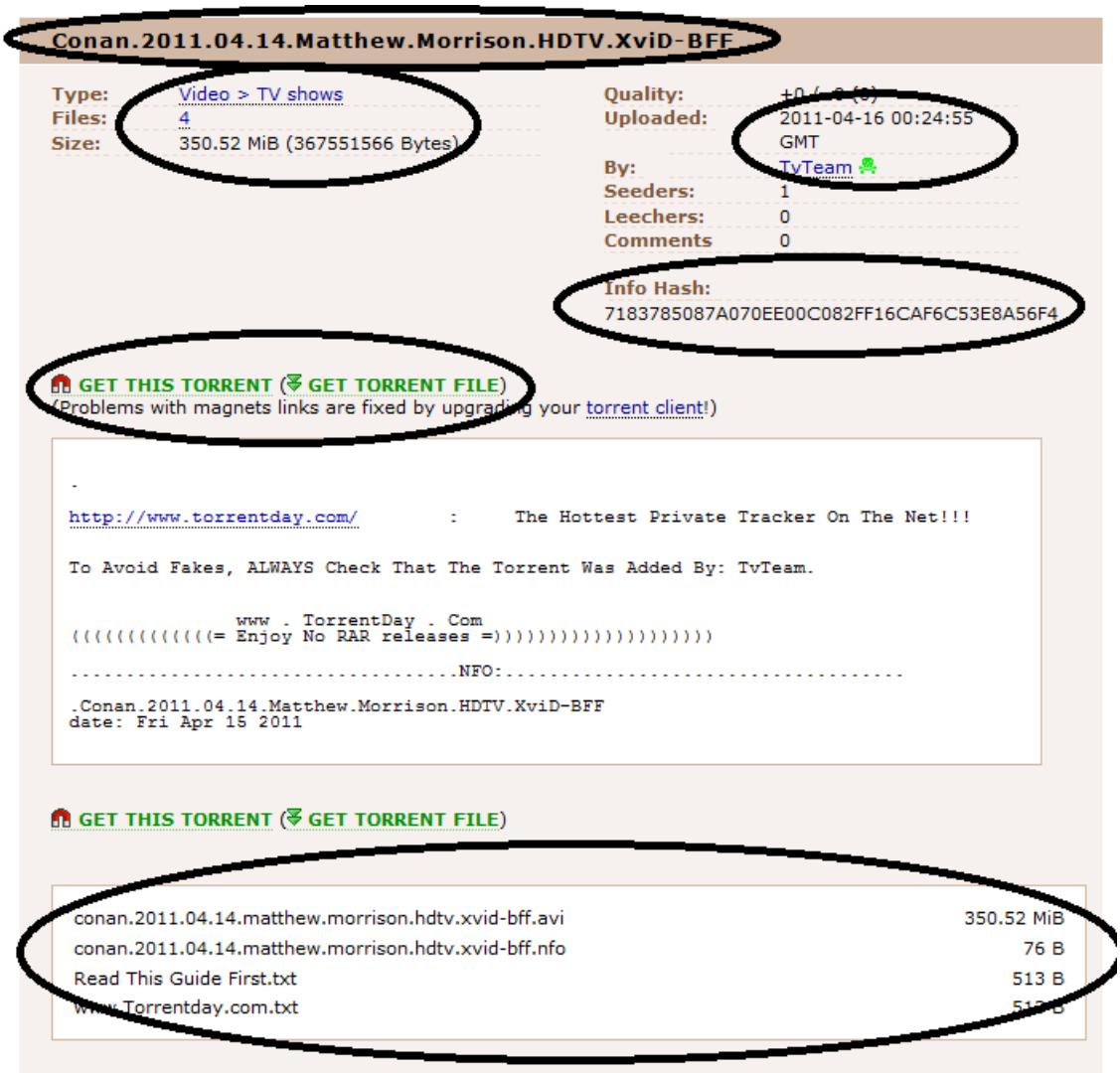

Figure 10: An example screenshot of the page from where the crawler fetches the data from.

7. **magnet** - The magnet link.
8. **publisher** - The username of the publisher.
9. **fileno** - The number of files related to the torrent.
10. **size** - The combined size of all the files in the torrent.
11. **uploadtime** - The time of upload of the torrent.

    The following items for each file related to the torrent.

12. **file** - The name of the file in the torrent.

13. **size** - The size of the file in the torrent.

The crawler inserted the data into two different tables in a SQLite database. The tables were named publish and files.

3.2 Database

We used the SQLite and MySQL databases in the managing of the data. There were some problems with the SQLite database. On some queries the following error was shown:

```
Error: database disk image is malformed
```

This error means that the database is corrupt somehow. We did not know where the problem was exactly, so we decided to export the sqlite database to a MySQL database.

There were two tables in the sql database, named files and publish. We used the .dump command to dump the contents of each table to a sql file separately like so:

```
sqlite>.output publish.sql
sqlite>.dump publish
sqlite>.output files.sql
sqlite>.dump files
```

We downloaded and installed MySql. We used the NetBeans IDE to connect to the MySQL server and created a new database called piratebay. We created the two tables using the SQL editor of NetBeans. We created the publish table using the following command:

```
CREATE TABLE publish(
            infohash VARCHAR(45),
            type TEXT,
            subtype TEXT,
            infourl TEXT,
            name TEXT,
            torrenturl TEXT,
            magnet TEXT,
            publisher TEXT,
            fileno TEXT,
            size TEXT,
            uploadtime TEXT,
        PRIMARY KEY(infohash ASC));
```

Then we created the files table with the command:

```
CREATE TABLE files(
            infohash VARCHAR(45),
            file VARCHAR(240),
            size TEXT,
          PRIMARY KEY(infohash ASC, file ASC)          FOREIGN
KEY(infohash) REFERENCES publish(infohash));
```

We used JAVA to create a simple application, to insert the data into the database. The application reads the contents of the SQL files and inserts them into the pirate-bay database. Each SQL file contains INSERT statements used to insert the data into the table.

The SQL files are read one line at a time and the line is checked to not contain any illegal characters. Also, the SQL syntax differs a little bit between sqlite and MySQL, so the SQL syntax is checked to be correct on each line. For instance, we had to remove all the " characters from each line. Some lines were incomplete, missing some data, so we ignored those lines. In the end, for the publish table, we were able to insert 651 712 entries out of 651 721 entries into the database. For the files table, we inserted 15 723 955 entries out of 15 723 962, so we had to remove only a few malformed entries.

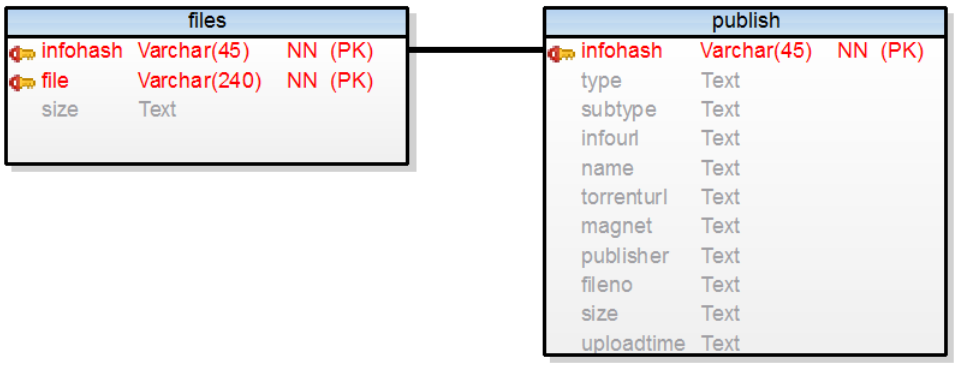

Figure 11: Database diagram for the piratebay database.

The database diagram for the database can be seen in figure 11. In this research, we had use only for the publish table. The files table was not used.

4 The Use of BitTorrent in the Pirate Bay Data Set

This section will provide the results obtained from the Pirate Bay data set described earlier. At first we will discuss the methods used in gaining the results. Second we will present the results for who the major publishers are, what is the content distribution and what are the incentives of the major publishers and compare them with the results from previous studies. Then we will provide deeper analysis on the data. Lastly we will draw conclusions from the results.

4.1 Research Methods

We used the collected data to gather useful information on the use of BitTorrent in the Pirate Bay website. We used SQL queries to fetch the data from the MySQL database. The SQL queries are described in detail in appendix A at the end of the document.

The results from the queries were saved in text files. The data in the text files was manipulated into wanted form using Java. Some of the graphs were made using

Microsoft Office Excel and some were made using Java and the JFreeChart library [17].

For the incentives of the major publishers we went to the Pirate Bay website [27] and found the description pages of the torrents uploaded by each major publisher. We checked to see what was in the description box and whether or not they were advertising some websites.

4.2 Major Publishers in the Pirate Bay Data Set

Using the data set on the torrents uploaded to the Pirate Bay during ten months of the year 2011, we identified the usernames who have uploaded the most torrents to the Pirate Bay web site. In the data set the biggest uploader is Anonymous. However this not a single user. Pirate Bay allows users to upload torrents anonymously and in this case the username of the publisher of the torrent is Anonymous. The number of anonymous uploads is 63 667 which is 9,77% of all of the uploads. The anonymous uploads are ignored.

There are four usernames who have uploaded clearly the biggest number of torrents to the Pirate Bay web site. Nine of the biggest publishers and the number of torrents uploaded can be seen in table 4. The biggest uploader has uploaded 75,1 torrents each day on average. The fifth biggest publisher has uploaded 8870 torrents and the 100th uploader has uploaded a bit more than 500 torrents, so the number of torrents published by users declines quite fast. However, this is still more than one torrent uploaded each day. As the number of distinct uploaders is 81042 and the 5000th publisher has uploaded only 12 torrents, it looks like most of the publishers upload only very few torrents.

Figure 12 shows how many torrents each user in the top 20 has uploaded. As can be seen, the number of torrents published by users declines very fast. The top publishers have published thousands of torrents in the eleven month period when most of the uploaders have uploaded only a few torrents.

What is surprising in the studies is that the number of torrents published by the top uploaders is so large compared to the other publishers. A very small number of usernames contribute a significant portion of the torrents. They are uploading tens of torrents a day so there must be a community of users behind the usernames of the biggest publishers. The people behind these usernames are obviously gaining profit from uploading so many torrents. Cuevas et al. [8] estimated the values of some

Username	Torrents Uploaded	Torrents/Day	Percent of All Torrents
TvTeam	23139	75,1	3,55%
scenebalance	21847	70,9	3,35%
sceneline	18134	58,9	2,78%
exmnova	14139	45,9	2,17%
Rabiner	8870	28,8	1,36%
chkm8te	7145	23,2	1,10%
hero0000	4090	13,3	0,63%
torrentzone	3636	11,8	0,56%
.BONE.	3617	11,7	0,56%

Table 4: Nine of the biggest uploaders of torrents to the Pirate Bay web site during the 308 days of data gathering.

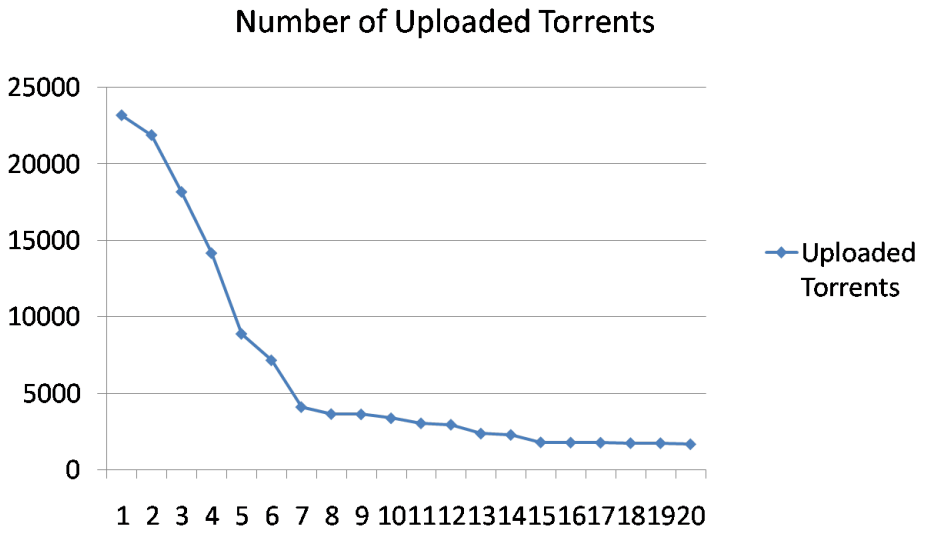

Figure 12: Graph shows the uploaded torrents during the 308 day period for the top 20 uploaders.

of the web sites promoted by the biggest publishers of torrents. Their estimates of the values of the web sites vary from tens of thousands of dollars to even millions of dollars. The number of visits to these web sites vary from tens of thousands of visits per day to hundreds of thousands visits per day.

4.3 Content Distributed in the Pirate Bay Data Set

We have data on 615 712 torrents uploaded to the Pirate Bay during the year 2011. The torrents are put into categories according to their content. The categories are Video, Audio, Porn, Applications, Games and Other. These categories differ a little from the categories used in the other researches. In the other research porn content was put into video category and software category included both applications and games. The data includes torrents that could be fake and have been removed from the Pirate Bay. The results are shown in figure 13.

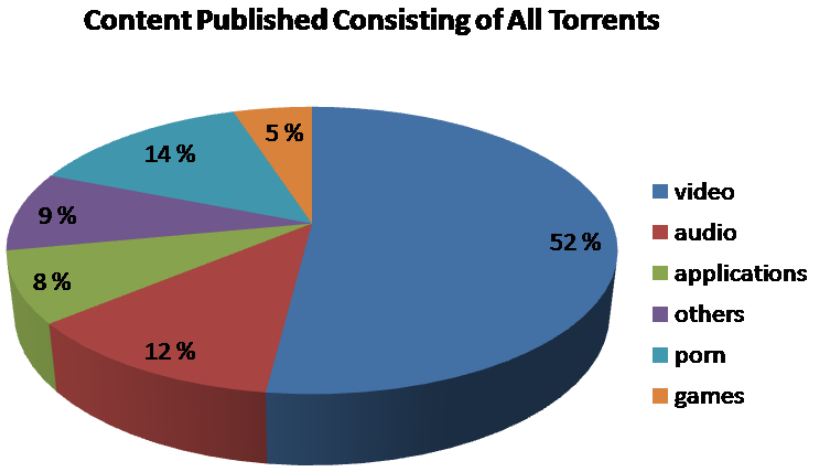

Figure 13: Percentage of torrents in each category for all of the torrents uploaded to the Pirate Bay during ten months of 2011.

As can be seen, the video category is pretty much the same size in both researches. Audio content is a few percentages smaller in the newer Pirate Bay data. Software category is about the same size as applications and games combined.

The results above show that BitTorrent is used in the distribution of all kinds of digital content from movies to e-books. The results in each of the studies indicate that video files contribute for the largest portion of files to be distributed using BitTorrent. This result is not surprising in the sense that video material is so easy

to produce. One can record material from the television, copy DVDs of movies or even make their own videos. Also the fact that BitTorrent is very well suited for the sharing of illegal and copyrighted content supports this. There is no central entity that can be held responsible for the distribution of the content. All the peers are responsible and removing one peer does not affect the overall distribution of the content.

Audio comes as the second largest content category across the studies, competing closely with the others category. The audio category includes, among others, music and audio books, music taking the largest portion. The content for this category is also easy to produce. A lot of people will transfer the contents of the their music CDs to the computer so they can listen to them on their mp3 players. Also, people like to listen to music and they might try new music first by downloading it for free before buying it, thus it is not surprising that audio is the second largest category.

The sizes of the rest of the categories, software and others, are close together across the different studies. Porn content takes a big chunk of the others category and its popularity does not require an explanation. Table 5 shows the differences of the results from the study done by Cuevast et al. [8] and the results obtained from the Pirate Bay data.

Category	Data from Cuevas et al. [8]	Pirate Bay Data
video	51%	52%
audio	18%	12%
software	17%	13%
others	14%	23%

Table 5: Comparison of content published according to Cuevast et al. and the Pirate Bay data.

Software and games contribute for a surprisingly low percentage of the torrents. The reason for this might be that the applications often require an activation code of some sort. This means that the software cannot be distributed without cracking the executables. An average person is not able to do this, so a random person who has bought an application will not publish it as it is of no use to others.

4.4 Incentives of the Big Publishers in the Pirate Bay Data Set

In this section we investigated the four usernames of the biggest uploaders of torrents presented in section 3.3.

The biggest uploader of torrents of all, TvTeam, seems to be concentrating on uploading torrents of movies and TV-shows. The username uploads on average as much as 75 torrents each day. In the description page of the torrents, the user is advertising at least two different private torrent sites. One of the sites is a movie torrent site named Torrenting (http://www.torrenting.com/) and the other one is called TorrentDay (http://www.torrentday.com/). These sites work by invite only. This means that one has to know someone who is already a member of the site to get an invite from them. Sometimes the sites offer free memberships, though. The sites promise fast download speeds, but often also require users to upkeep a large enough upload/download ratio. These private sites have their own private trackers, which require users to be logged in to be able to connect to the tracker. This way they can track the amount of content the users are uploading and downloading.

Another way to get an account in one of these sites is by donating money to them. This allows users to get VIP accounts and maybe allows them to invite their friends to the site also. It seems like the donations are the main way for these sites to make money. As of writing this one of TvTeam's website is offering free memberships and the other is invite only.

The second biggest uploader is the username scenebalance. Scenebalance is also concentrating on uploading torrents of movies. Scenebalance is advertising a private torrent community on the description pages of the torrents. The site is called Scenetime (http://www.scenetime.com/). The site is invite only, with the possibility of getting a VIP account with 4 GB of upload credit and one invite for a $7 donation.

The third biggest uploader, sceneline, is publishing torrents of tv-shows. Sceneline is advertising a private torrent site called SPEED.CD (http://speed.cd/). The web site is invite only, but with a donation of $10 it is possible to get a VIP account.

The fourth username we investigated is exmnova. Exmnova is uploading torrents of porn content. The username is also advertising a private torrent site. This site is focused entirely on porn content and it is called 69Bits.com (http://www.69bits.com/). Currently they have more than 50,000 registered users and the site is offering free

memberships.

All of the four usernames are VIP or trusted members of the Pirate Bay. This means that the users do not upload fake content and other users are happy with their releases. Most of the content uploaded is copyrighted and this is one of the reason for their popularity.

In conclusion, it seems like uploading torrents to the Pirate Bay and advertising their own sites in the description pages of the torrents is profitable. Looks like they are gaining enough income from the advertisements and the donations made to their sites to make this profitable.

4.5 File Size Distributions

In this section we present the file size distributions for the torrents uploaded to the Pirate Bay during the monitoring period. We calculated how the sizes of the files are distributed among all of the files and among the different types of files. To get the file size distributions we counted the number of files for each megabyte. For each megabyte we added the number of files in the previous megabyte to the number to get the total number of files. Figures 14 and 15 show how many of the files are of the shown size or less for all of the files and for the different categories.

Figure 14 shows the file size distributions for all of the files and the video files. The figure includes all of the files with size less than or equal to 7 GB. The total number of files is 651 708, so only a small number of the files are bigger than 7 GB and thus not shown on the figure. Figure 14 also presents the size distribution for the video files. The total number of video files in the data is 338 115, so only a few of the files are not shown in the figure. This is done so the graph would be more readable.

As can be seen in figure 14 there are some considerable fluctuations in the graphs. There are actually more than 15 000 files with size of only a few bytes. These are usually test torrents or some random text files.

The graph for video files takes its first jump at around 175 MB. This is probably so, because a 21 minute tv-show episode encoded as an avi is often around 175 MB in size. The next jump is at the 350 MB mark which is the size of a 42 minute episode. There are a lot tv-shows being published so these jumps seem rational. A bigger jump in the graph is at around the 750 MB mark. This is the size of most of the CDs so anyone wanting to save the file on a CD can do so. A 42 minute tv-show in 720p resolution is frequently about this size. Also a two hour movie is around this

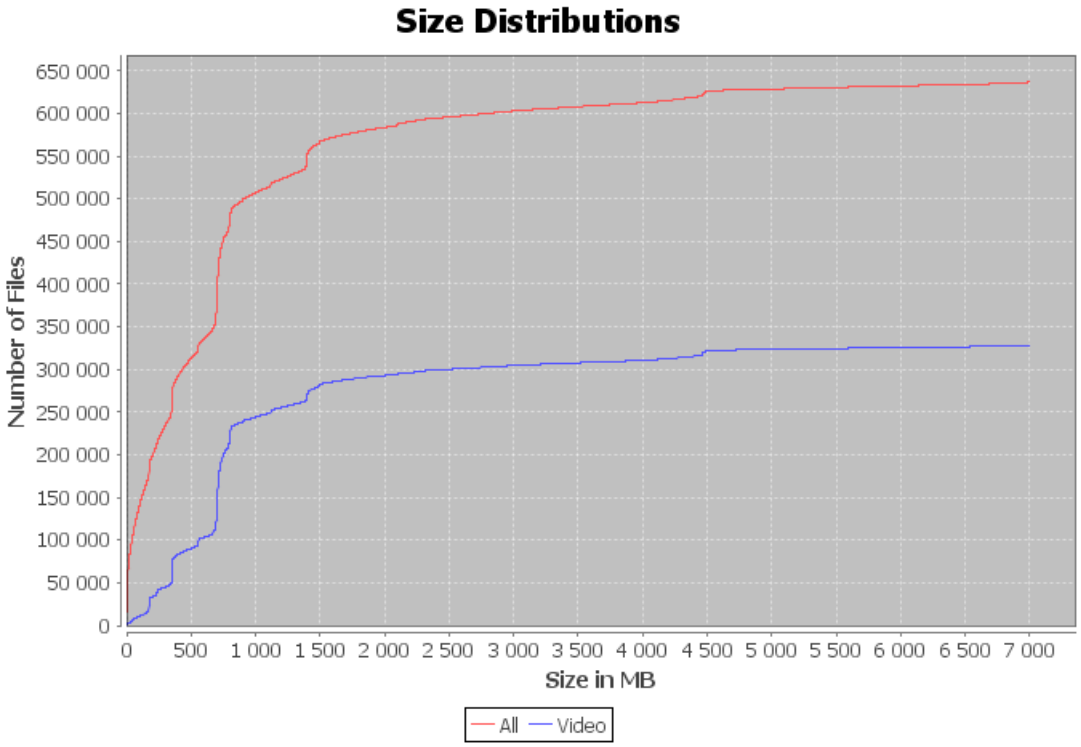

Figure 14: Graph shows the file size distributions for all of the files and video files.

size. A smaller jump can be seen around the 1.4 GB mark. This could be, because movies encoded in 720p resolution are commonly this size. The final leap in the graph can be seen at 4.5 GB. DVDs are about this size so these files can be saved on a DVD. Movies with good picture quality are often made this size. The graph for all of the published files accompanies the video graph. This is because more than half of all of the files published are video files.

Figure 15 shows the file size distributions for the rest of the categories, named, Audio, Other, Porn, Applications and Games. The total number of files in each category are presented in table 3 on page 10, thus only a small number of files is excluded from the graph. However, for the games category the line still goes gradually up, as there are some games larger than 3 500 MB in size. All of the graphs move up quite fast, meaning that a large percentage of the files are quite small. In the Audio category the graph rises steadily and starts to even out at the 500 MB mark. This could be, because people publish everything from single songs to full albums to discographies and all of these can vary in size.

The category, named Other, starts quite high on the plot, because the test and txt files are put into this category. Their size is usually only a few bytes. Files are often

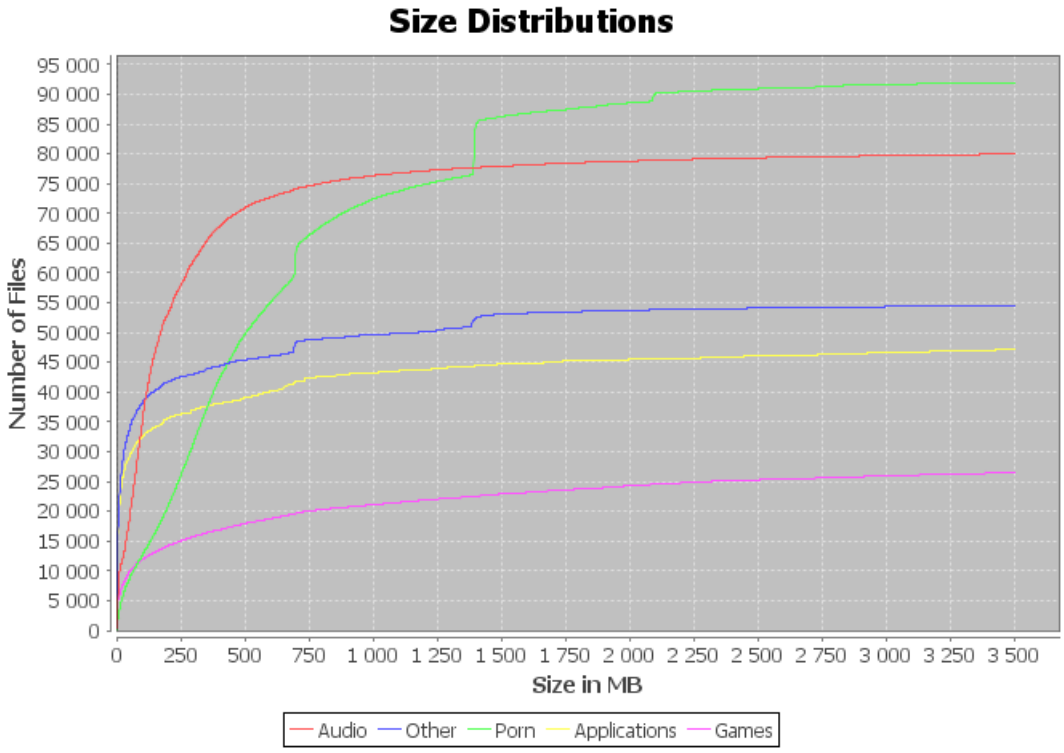

Figure 15: Graph shows the file size distributions for audio, porn, application, game and other files.

misplaced into this category, which might explaing the small jumps at the 740 MB and 1.4 GB marks. This may be caused by video files being put into this category by mistake.

Porn files gradually increase in size. This might be caused by different sized clips. There are jumps at around the 740 MB and 1.4 GB marks. This could be caused by the same reasons mentioned about the video files.

Most of the applications seem to be very small as are the published games. However, relatively larger number of games are bigger in size than the applications.

All in all, about 85% of the files published seem to be less than 1.5 GB in size. This information is useful in determining the optimal piece size for the torrent files.

4.6 Publish Activity of Content Types

This section provides information on the publish activity of the different types of content published in the Pirate Bay during the monitoring period. We provide graphs for the number of uploads per day for different types and four of the biggest

publishers. We also counted the Pearson product-moment correlation coefficients and the relevant t-values for the data. Lastly, we show the rate at which torrents are uploaded to the Pirate Bay.

Figure 16 shows the publishing activity for the different types of content. The figure shows the number of uploads for each day of the monitoring period. Video is clearly the most uploaded type of content with the rest of the categories being quite equal. If all of the of the users uploading content acted as independent variables, the graphs should be quite level, since there is a large number of users. However, looking at figure 16 we can see that there are a lot of fluctuations in the publishing activity. The fluctuations are quite significant for each category. Some of these fluctuations might be caused by down time of the Pirate Bay website.

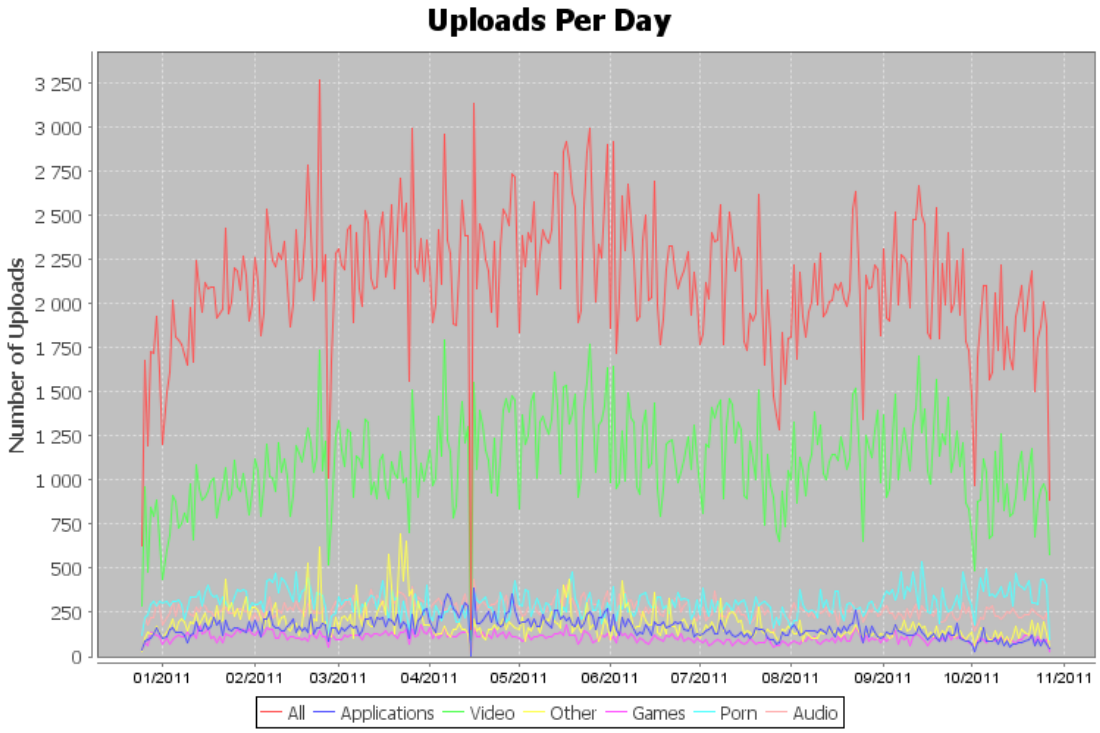

Figure 16: Graph shows the publish activity of the different types for each day of the monitoring period.

There does not seem to be much of a difference in the publishing activity between different times of the year, but daily fluctuations are very high. There are huge fluctuations in the publishing patterns of the top four publishers, as can be seen in figure 17. A large portion of the uploads are done by these users. This means that fluctuations in the publishing patterns of the biggest uploaders will reflect on the publishing patterns of all of the torrents.

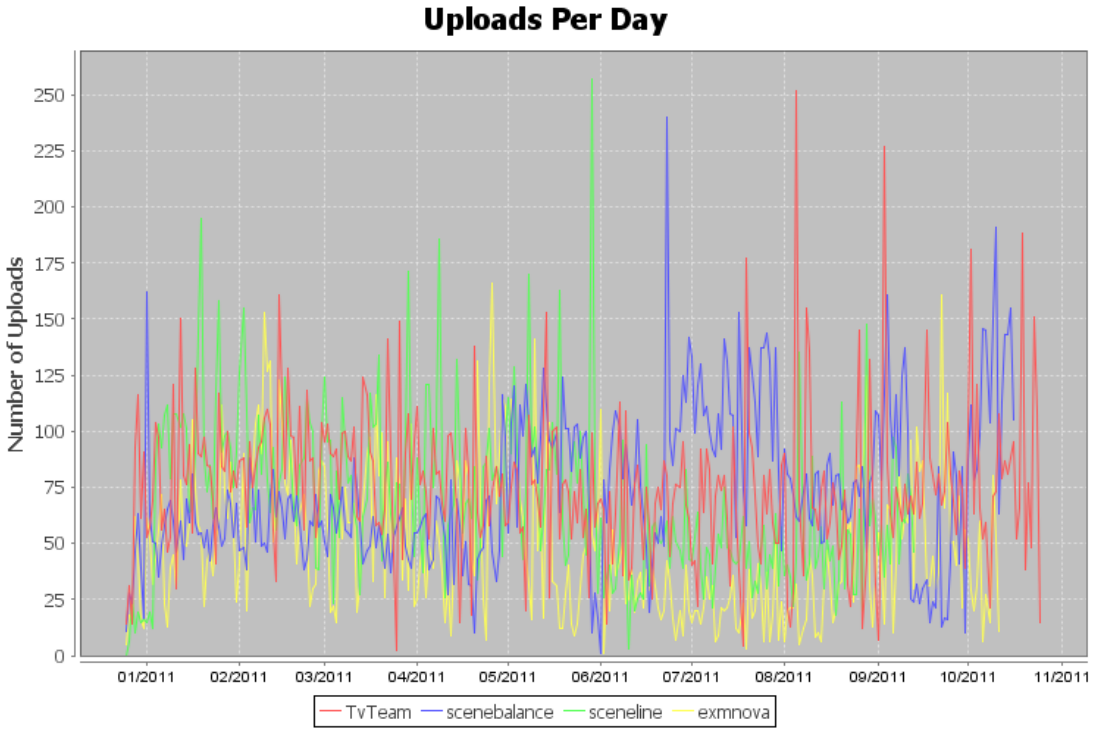

Figure 17: Graph shows the publish activity of the top four publishers for each day of the monitoring period.

The top usernames are publishing torrents so much that there must be groups of people behind the usernames. The fluctuations in the publishing activity could be caused by a different number of torrents being available for uploading on different days. For example, tv-shows come out on a particular day of the week, so publishing a torrent of it probably happens on the same day of the week.

It also may be that the publishing activity of regular users, who upload only a very small amount of torrents, is not totally random. They might upload torrents during the weekends, when they have free time. On the other hand, the organizations behind the big publishers, who seem to be uploading torrents as their jobs, upload more during the week.

The number of uploads for all of the content published each day of the week can be seen in figure 18. Tuesday and Wednesday are the busiest days and Sunday is the quietest day regarding the number of uploads.

We calculated the Pearson product-moment correlation coefficients for the different types of uploads using the upload data for each day. The correlation coefficients were calculated using the formula:

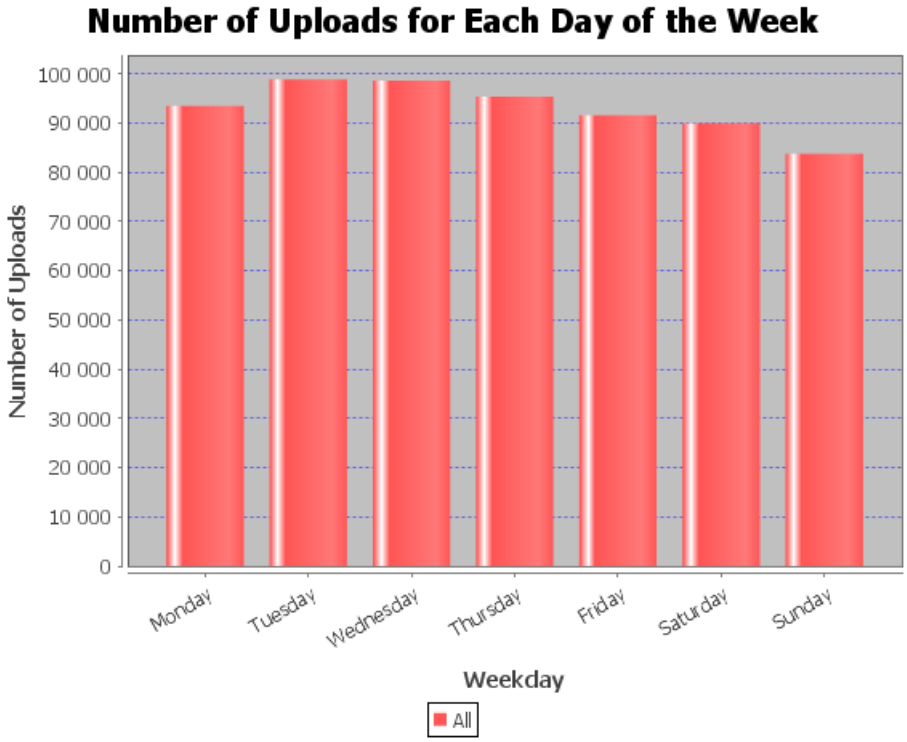

Figure 18: Graph shows the publish activity for each day of the week.

$$r = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

where X_i and Y_i go over the values of samples X and Y and $\overline{X}$ and $\overline{Y}$ are the sample means respectively.

There seems to be some correlation among the daily publish activity of different types of content, as can be seen in table 6.

We also tested the significances of the correlation coefficients. We counted the t value to test the significance by using the following formula:

$$t = r * \sqrt{(n-2)/1 - (r^2)}$$

where r is the correlation coefficient and n is the number of values in the dataset.

The relevant t values to test the significances of the correlation coefficients can be found in table 7. The table 8 provides the limit t-values for the probabilities for the two degrees of freedom, 22 and infinity. The degree of freedom is the number of items in the data minus two.

The t-values in table 7 show that the probability of there being correlation is very high, in most cases. According to table 8, there is more than a 99,9% probability that there is correlation among the daily publishing patters for most of the types of

content.

	All	audio	video	games	applications	other	porn
All	x	0.667	0.905	0.603	0.658	0.527	0.388
audio		x	0.455	0.548	0.582	0.333	0.255
video			x	0.385	0.512	0.265	0.195
games				x	0.556	0.390	0.261
applications					x	0.295	0.019
other						x	0.101

Table 6: The correlation coefficients for the different types of uploads per day datasets.

These correlations among the different types are likely caused by the fluctuations in the publishing patterns. Also, according to figure 18, the publish activity is a bit different each day of the week, which might explain some of the correlation between the different types of content.

	All	audio	video	games	applications	other	porn
All	x	15.658	37.141	13.218	15.280	10.846	7.368
audio		x	8.930	11.468	12.526	6.174	4.604
video			x	7.296	10.418	4.798	3.480
games				x	11.705	7.417	4.722
applications					x	5.397	0.330
other						x	1.783

Table 7: The t values to test the significance of the correlation coefficients for the different types of uploads per day datasets. The relevant degree of freedom is infinity.

One Sided	90%	95%	97.5%	99%	99.5%	99.75%	99.9%	99.95%
Two Sided	80%	90%	95%	98%	99%	99.5%	99.8%	99.9%
22	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
infinity	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

Table 8: The t-distribution table for degrees of freedom 22 and infinity.

Figure 19 shows the number of uploads per each time interval, where each upload is the time between two consecutive uploads. Each time interval is one second long. The first bar is the time interval from zero to one seconds, the next, the highest bar,

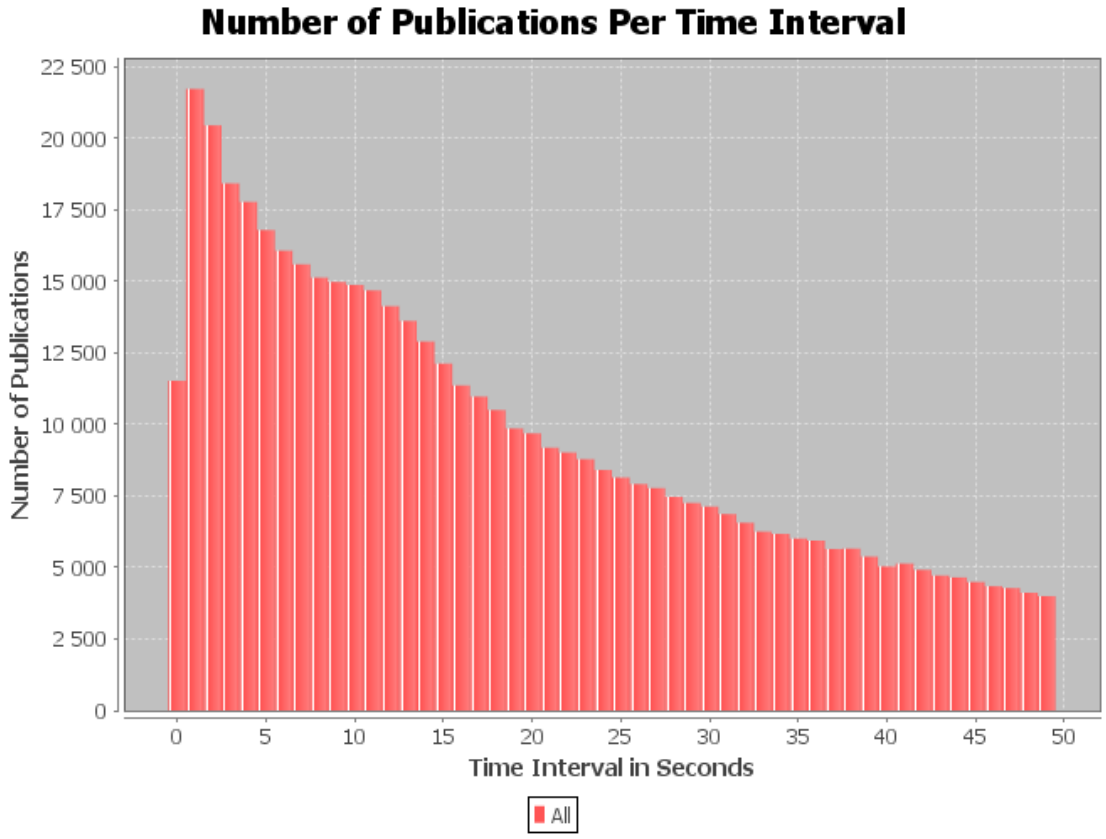

Figure 19: Graph shows how fast content is published on the Pirate Bay. The first bar is the time interval 0-1 seconds. The second bar is the time interval 1-2 seconds and so on.

is the time interval from one to two seconds and so on. So, if the time between two uploads is less than one second, it goes into the first interval. If the time between two uploads is one to two seconds, it goes into the second time interval and so on.

According to figure 19, the most usual rate at which content is published on the Pirate Bay is once every 1 to 2 seconds. This is about about 3.4% of the torrents. Only about 1.8% of torrents are uploaded at a faster rate. This is important information for the admins of the servers, as they have to make sure that the system is able to handle this many interactions.

From this peak at 1-2 seconds, the rate at which uploads happen goes steadily down, except for a small jump at 9-14 seconds. This is because one of the top publishers, exmnova, seems to upload most of its content at this rate, as can be seen later figure 29.

In summary, the publish activity of the torrents on the Pirate Bay portal varies a

lot. These deviations in the publishing activity may be caused by down time of the Pirate Bay website, fluctuations in the publishing activity of the top publishers, national holidays or weekdays. One would think that the publishing activity with so many unique users would be quite level, but surprisingly this is not the case.

4.7 Top Publishers' Publish Activity

In this section we took a closer look at the publish activity of the publishers who have uploaded the most content into the Pirate Bay during the monitoring period. We checked to see if there was a difference in the hourly or the daily publish activities of the top publishers. Also, we analyzed the rate at which content is published in the Pirate Bay by the top publishers.

Figure 20 shows the number of uploads for each of the day. The times are in Pirate Bay's timezone, which is the Central European Time (UTC + 01:00). There seems to be a small apex in the uploads at the 11 hour mark, but other than that the number of uploads during each hour is quite equal. One could think that most users would upload torrents in the evening, after work. This would mean that most of the users of the Pirate Bay would come from around the timezone of UTC+8, which is used in central Russia and China, among others.

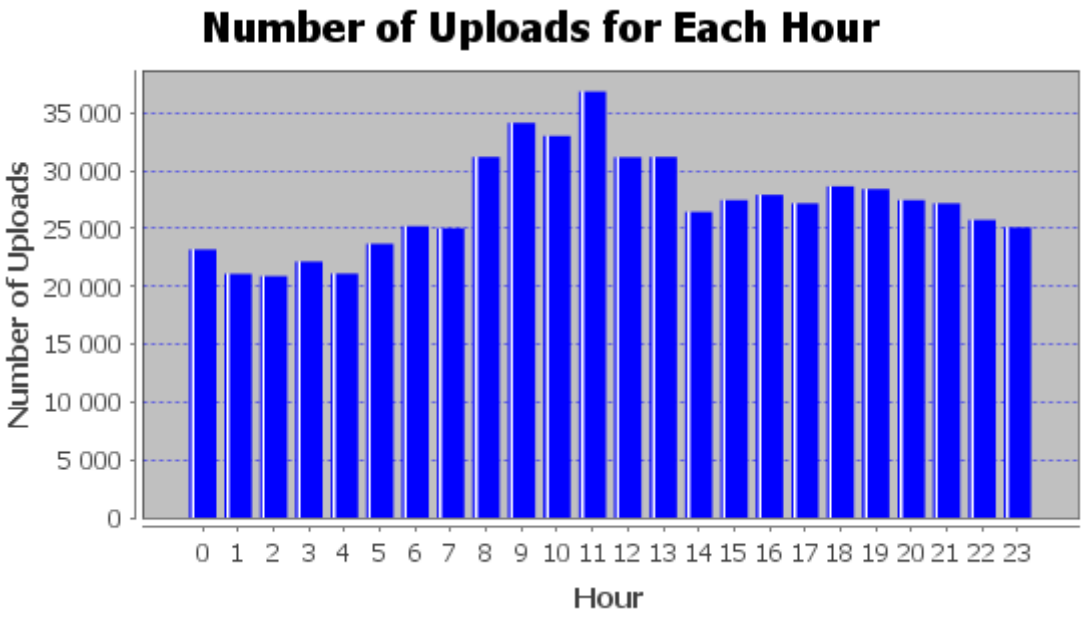

Figure 20: Graph shows the number of uploads for each hour of the day.

The figure 21 shows the number of uploads for the three biggest publishers for each

hour of the day. As can be seen in the figure, the hourly publishing pattern for the three top publishers, TvTeam, sceneline and scenebalance, looks very similar. Their publishing activity goes up at around 22:00 and goes even higher during the early hours of the day. The highest number of published torrents for each user is at 5 in the morning. After that the publishing activity goes down, being the lowest at 13:00.

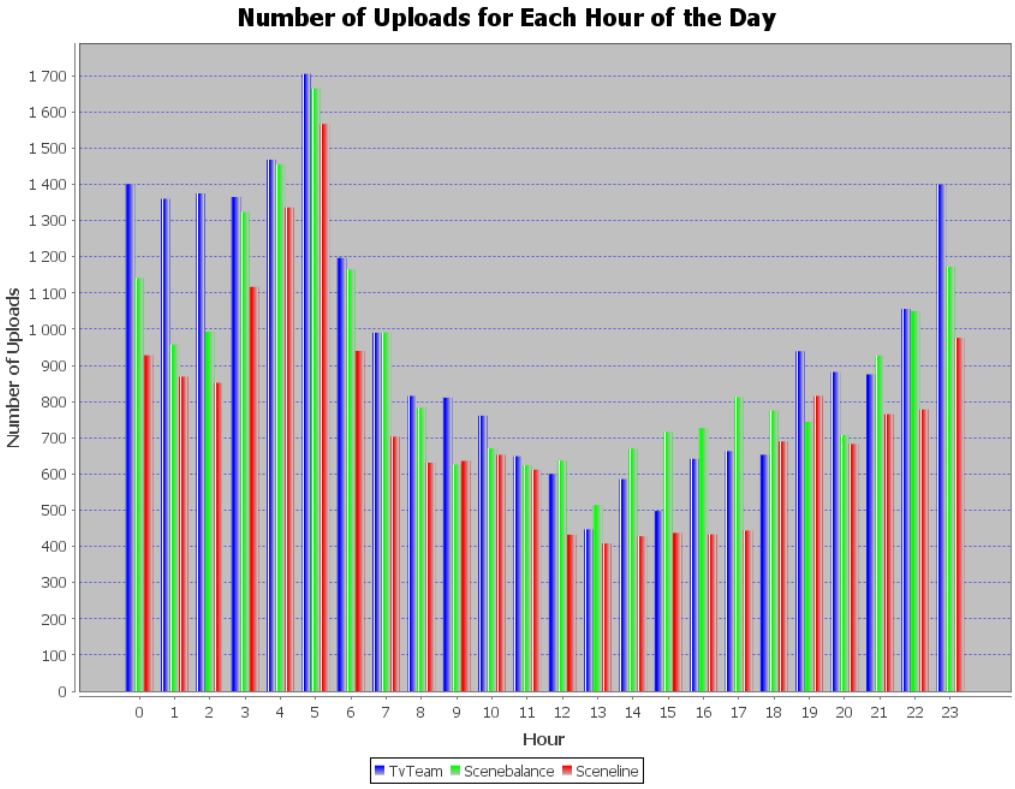

Figure 21: Graph shows the publish activity of the top 3 publishers for each hour of the day.

The top publishers contribute for a big percentage of the torrents, so their impact is considerable. As can be seen in figure 21, the three biggest uploaders seem to belong to the same timezone. The number of uploads for them seems to go up at 20:00 UTC+1:00 peaking at 5:00 UTC+1:00. If they are uploading torrents as their job and working during the day, it would mean that they belong to the timezones of UTC+8:00 to UTC+11:00, which is central Russia, China and Australia.

We counted the Pearson product-moment correlation coefficients for the data sets of uploads per hour for all of the uploads, anonymous uploads and uploads by the three top publishers. The correlations can be seen in the table 9. There seems to be a very high correlation between the three top publishers.

	All	Anonymous	TvTeam	scenebalance	sceneline	exmnova
All	x	0.895	-0.742	-0.741	-0.598	0.725
Anonymous		x	-0.564	-0.623	-0.488	0.596
TvTeam			x	0.902	0.920	-0.576
scenebalance				x	0.927	-0.594
sceneline					x	-0.420

Table 9: The correlation coefficients for the uploads per hour datasets.

The relevant t values can be seen in the table 10. The t values for the top three publishers are all more than 9.8. Looking at the critical points seen in table 8 for degree of freedom 22 (df = 24-2), we can see that there is more than a 99.9% chance that there is some correlation between the datasets for the top three publishers.

	All	Anonymous	TvTeam	scenebalance	sceneline	exmnova
All.	x	9.178	-5.188	-5.169	-3.504	4.936
Anonymous		x	-3.202	-3.734	-2.621	3.486
TvTeam			x	9.804	10.989	-3.303
scenebalance				x	11.621	-3.459
sceneline					x	-2.172

Table 10: The t values used to test the significance of the correlation coefficients for the uploads per hour datasets.

The correlation coefficients of the three biggest uploaders, TvTeam, scenebalance, and sceneline are all over 0.9. This is a very high correlation. Also the t-values for the correlations, shown in table 10, are very high, meaning that there is a higher than 99.9% probability that the correlation is there. This correlation is very likely caused by the groups of people behind the usernames working in the same timezone.

However, the fourth biggest publisher, exmnova, has a negative correlation coefficient of around -0.58 with the three biggest publishers. This suggests that exmnova is working in an opposite timezone. The t-values are still big enough to suggest that the correlation is very likely to be there.

We also counted the correlation coefficients between the top four publishers for their publish activity on each day of the monitoring period. Here the correlation seems to be quite small between the top publishers. The correlations can be seen in table 11. One thing to note is that sceneline has not uploaded anything until the 6th

of February, 2011. The respective t values for the correlation coefficients for the publish activity of each day data sets can be seen in table 12. Here the relevant degree of freedom is infinity.

The correlations are much smaller than for the hourly datasets. Most of the t-values are still large enough to refer to that there is a correlation. This suggests that although the users might belong to the same timezone their publishing patters differ greatly. In some cases the t-values are smaller though, meaning that there is a bigger probability of the correlation being caused by chance.

	All	Anonymous	TvTeam	scenebalance	sceneline	exmnova
All	x	0.559	0.183	0.173	0.447	0.331
Anonymous		x	0.185	-0.093	0.282	0.246
TvTeam			x	0.098	0.146	0.218
scenebalance				x	0.248	-0.107
sceneline					x	0.075

Table 11: The correlation coefficients for the uploads per day datasets.

	All	Anonymous	TvTeam	scenebalance	sceneline	exmnova
All	x	11.776	3.260	3.087	8.740	6.141
Anonymous		x	3.301	-1.631	5.136	4.441
TvTeam			x	1.720	2.587	3.907
scenebalance				x	4.478	-1.885
sceneline					x	1.322

Table 12: The t values used to test the significance of the correlation coefficients for the uploads per day datasets.

The small correlations among the users could be explained by the similar publishing pattern when only looking at the number of uploads per day of the week. This pattern is shown in figure 22. For the three biggest publishers Tuesday and Wednesday are the busiest days and Sunday seems to be the quietest day for all. For the fourth biggest publisher, exmnova, Saturday seems to be the day with the largest number of uploads. The correlation could be affected by worldwide holidays also.

We calculated the number of uploads for each time interval before in figure 19, where each upload represents the time between two consecutive uploads. In figure 23 we added the number of uploads from the previous time interval second to the number

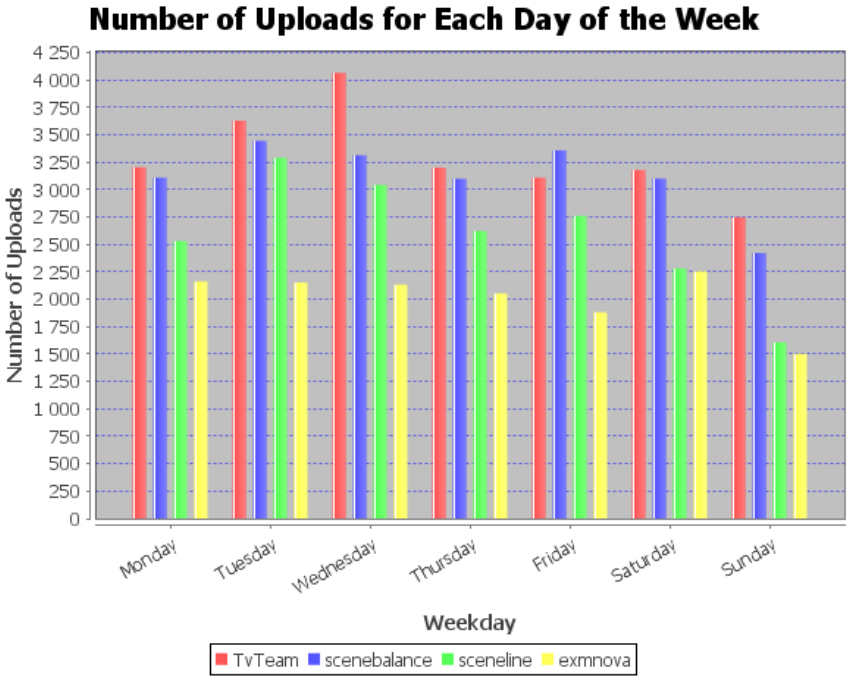

Figure 22: Graph shows the publish activity by the top four publishers for each day of the week.

of uploads in the next second. This way we got a graph showing how many of the time intervals between two consecutive uploads are equal to or less than the time shown in seconds.

Figure 23 shows that from the total of 651 712 uploaded torrents, about 600 000 consecutive uploads, or about 92% of the uploads, have happened within 100 seconds or less from each other. This means that torrents are being constantly uploaded to the Pirate Bay portal and any down time will affect the number of uploads significantly. Pretty much all of the consecutive torrents are uploaded within 250 seconds from each other, so there are not many big gaps in the uploading of the torrents.

In figures 24 and 25 we did the same for the top nine publishers as we did in figure 23 for all of the uploads. The graphs provide information on the rate at which the different publishers upload torrents. Figure 24 provides data up to thirty minutes and figure 25 provides the same data, but only up to two minutes enabling us to take a closer look at the data.

The uploading behaviour of the top nine publishers seen in figures 24 and figure 25 seems pretty surprising. It looks like most of the torrents are uploaded in a very

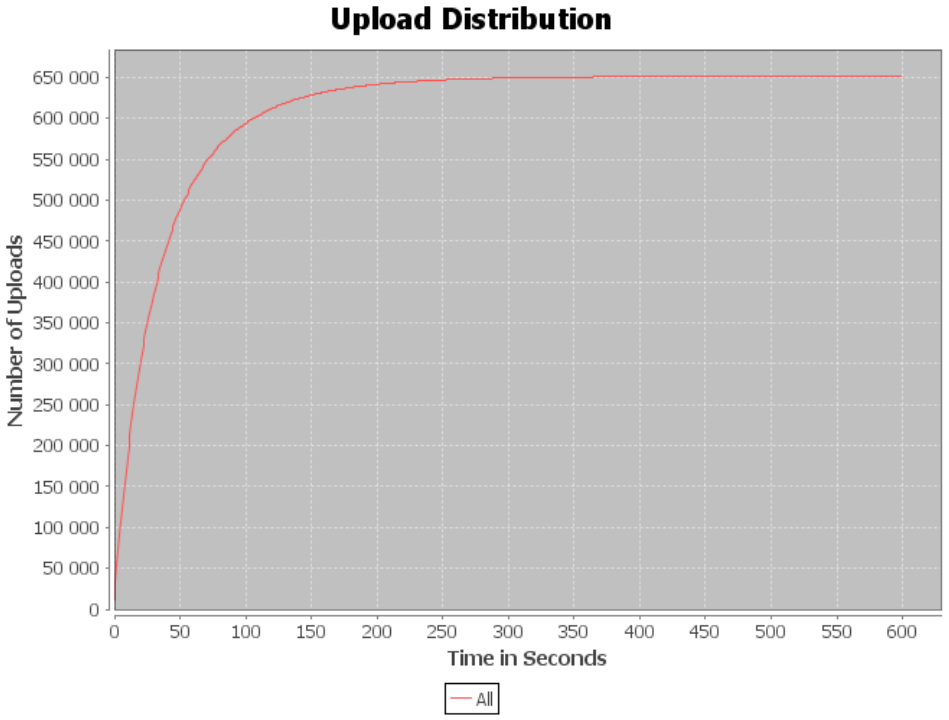

Figure 23: Upload distribution for all of the publications up to ten minutes.

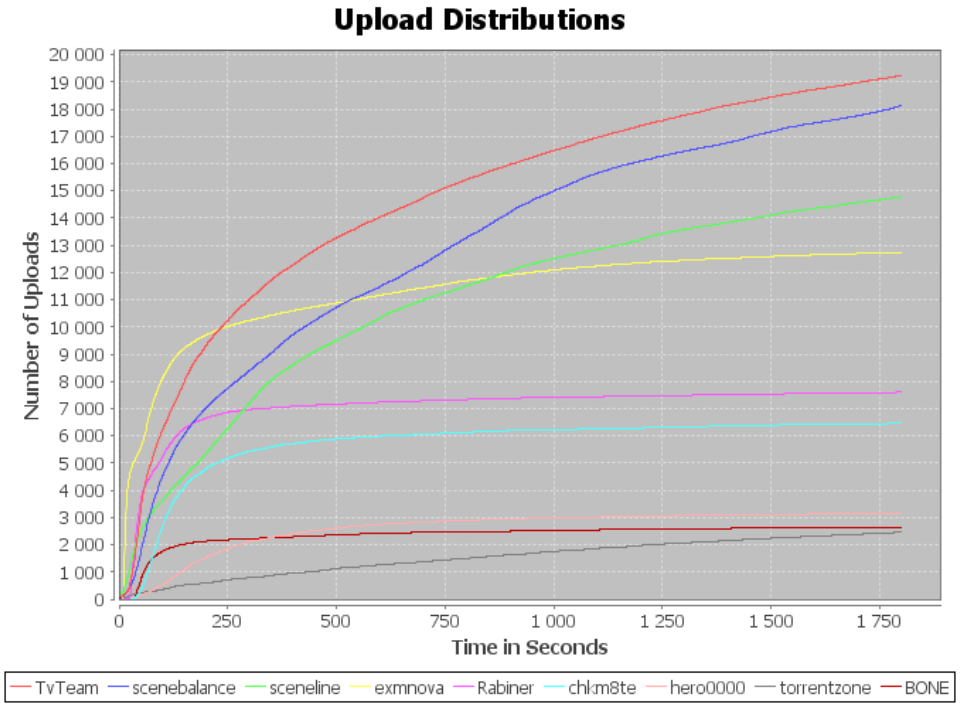

Figure 24: Upload distributions for the top nine publishers with data up to thirty minutes.

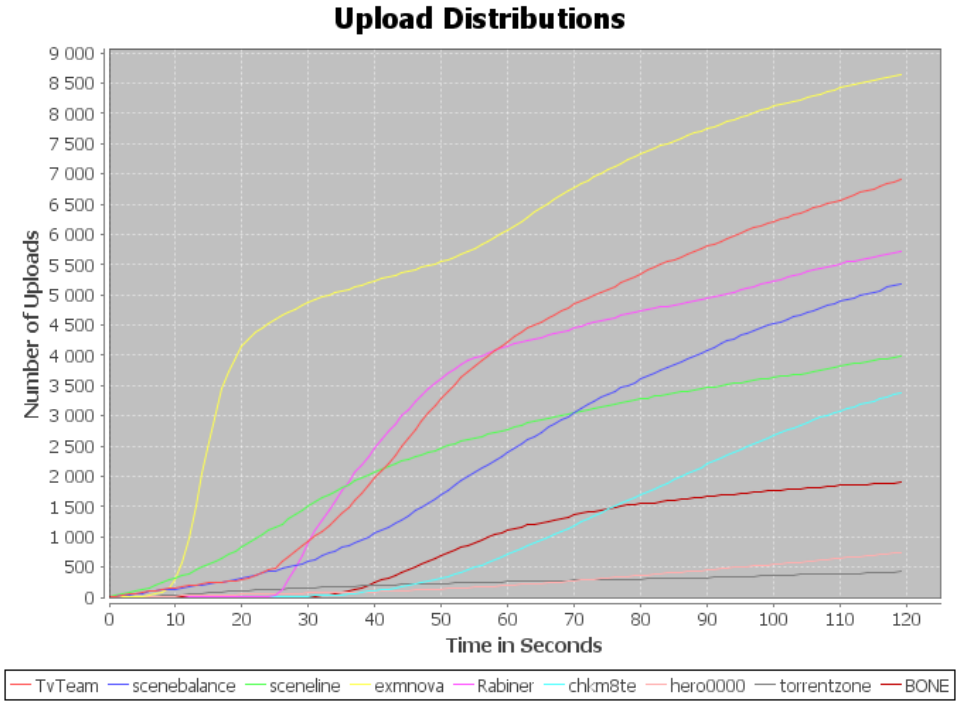

Figure 25: Upload distributions for the top nine publishers with data up to two minutes.

short time frame from each other.

Figures 26 - 30 demonstrate this very well. The figures provide similar data of the five top uploaders as figure 19 did for all of the torrents combined

For instance, the fourth biggest uploader, exmnova, has uploaded most of its torrents within 14-16 seconds. Assuming that only one user can be logged in with the username at the same time suggests that this user is using a script of some kind to upload the torrents. Maintaining an upload frequency this high by hand is next to impossible.

The publish patterns for all of the top publishers are a bit different. However, all of the publishers do have a clear peak in the graphs which could mean that they always work in a similar manner.

Also the other top publishers have uploaded the torrents at a very high frequency. The upload rate differs from user to user, but for all of the users the torrents must have been prepared beforehand and then the torrents are uploaded in chunks. Then the group of people behind the username prepares another batch of torrents and uploads them to the website. This behaviour is clearer for some of the usernames. For some usernames the times between uploads are more spread out than for others.

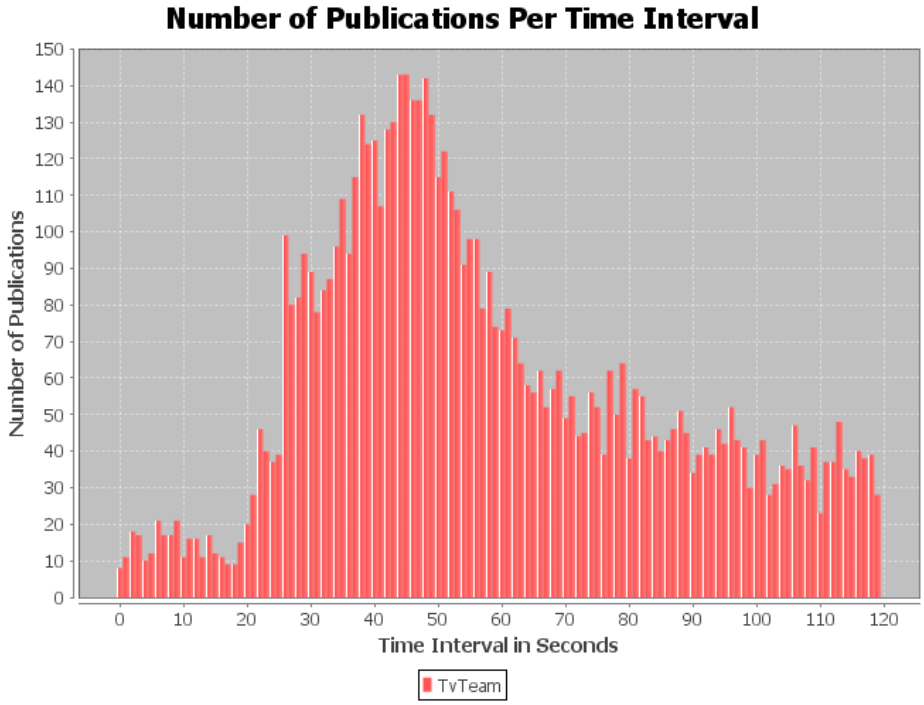

Figure 26: Graph shows the number of uploads for each time interval for the user TvTeam.

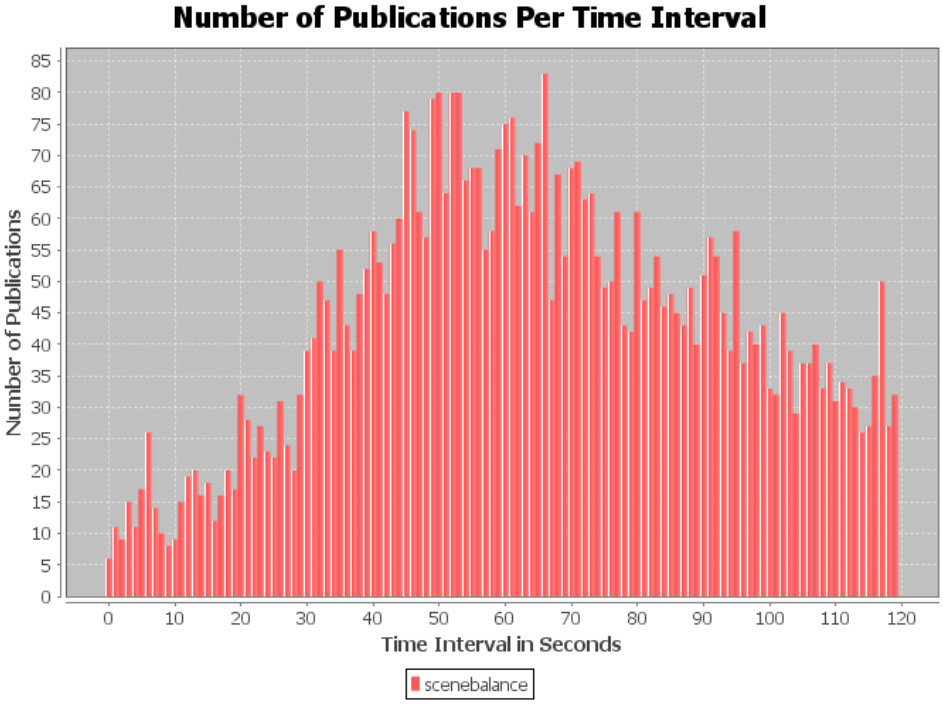

Figure 27: Graph shows the number of uploads for each time interval for the username scenebalance.

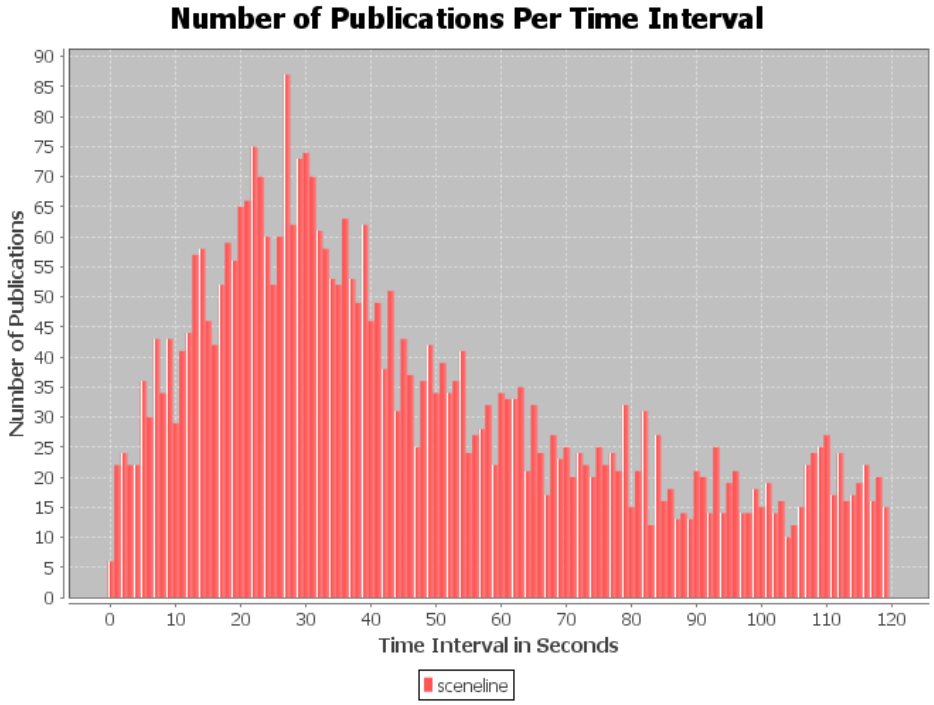

Figure 28: Graph shows the number of uploads for each time interval for the username sceneline.

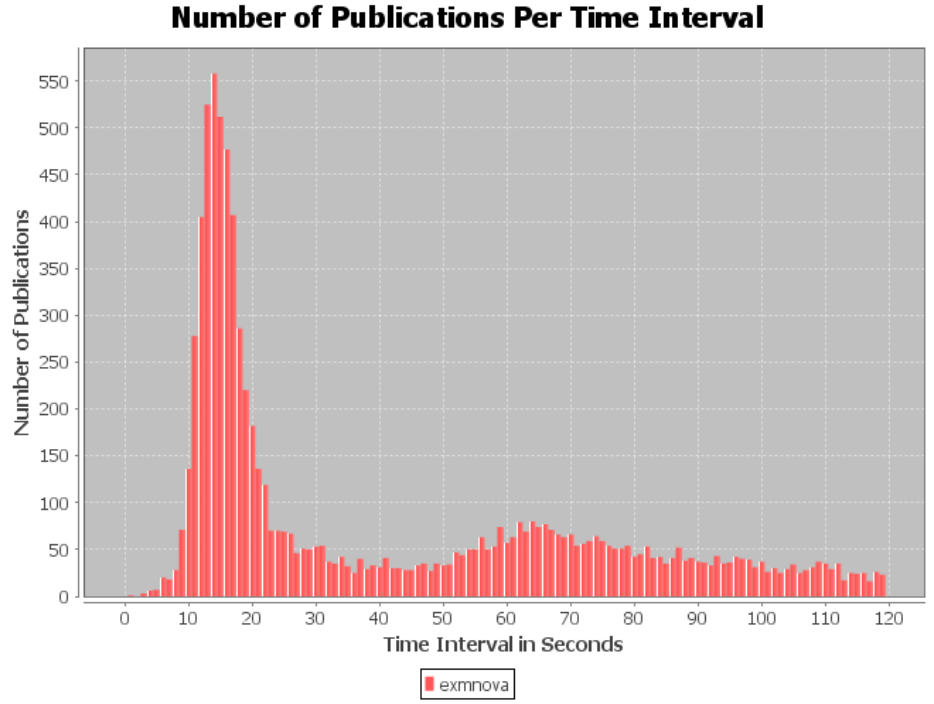

Figure 29: Graph shows the number of uploads for each time interval for the username exmnova.

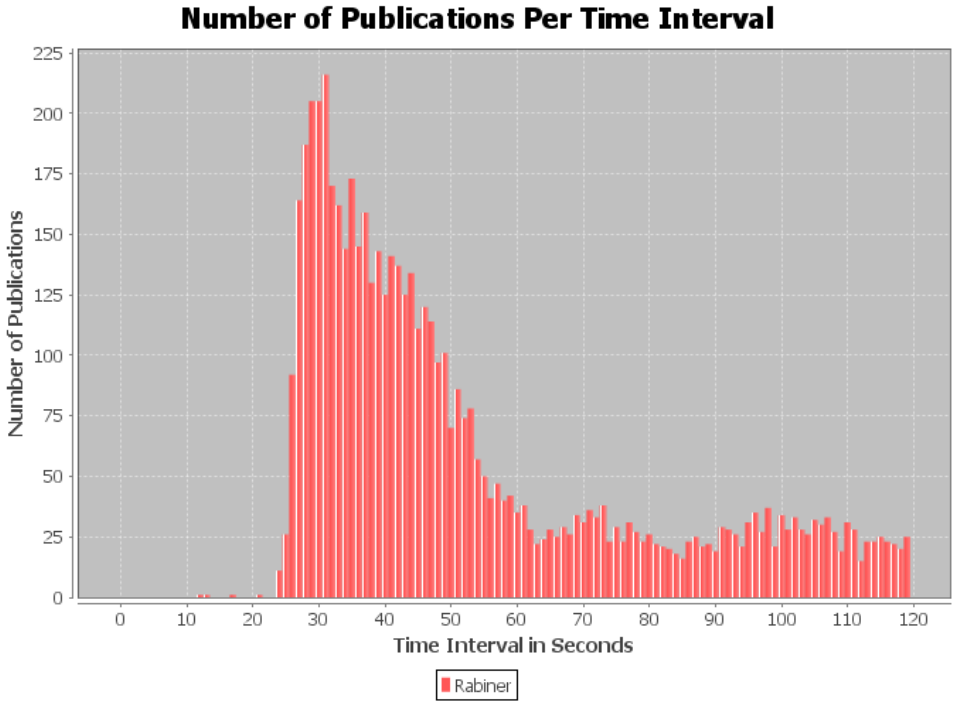

Figure 30: Graph shows the number of uploads for each time interval for the username Rabiner.

In conclusion, it looks like the Pirate Bay is in quite heavy use. Torrents are being added at a high rate and any downtime in the portal will affect this rate. Also, it seems like the behaviours of the top publishers are quite similar. The only real difference comes from the users being in different timezones. All of the biggest publishers prepare the torrents beforehand and upload them in chunks.

5 The Publishing of Movies in the Pirate Bay

The Pirate Bay can also be exploited in distributing copyrighted content. In this section, we will investigate the publishing speed of torrents of new Hollywood movies on the Pirate Bay web site. In the first subsection, we will describe the methods used in finding the publishing speed of the torrents and in the second section we will present the results.

5.1 Research Methods

We researched ten popular feature films released in 2011. The release dates of the movies were compared with the upload times of the torrents on the Pirate Bay. The criteria for selecting the movies were that they were popular enough to be shown in movie theaters and later on they were released on DVD or Blu-ray. The release dates for both the premiere and the DVD had to be between the 25th of December, 2010 and 28th of October, 2011, because that is the time interval of the data we have. The selected movies are shown in table 13.

Movie title	Release date	DVD/BR release date
X-Men First Class	1.6.2011	9.9.2011
Harry Potter and the Deathly Hallows: Part 2	15.7.2011	11.11.2011
Midnight in Paris	10.6.2011	20.12.2011
Transformers: Dark of the Moon	29.6.2011	30.9.2011
Pirates of the Caribbean: On Stranger Tides	18.5.2011	18.10.2011
Source Code	1.4.2011	26.7.2011
Sucker Punch	25.3.2011	28.6.2011
Gnomeo&Juliet	10.2.2011	24.5.2011
Bridesmaids	13.5.2011	20.9.2011
Sanctum	3.2.2011	7.6.2011

Table 13: The ten selected movies and their release dates.

We searched the Internet Movie Database (IMDB) to find the ten movies and their release dates [15]. Eight of the ten selected movies were in the top 100 of the most popular feature films released in 2011 according to IMDB's MOVIEmeter. The MOVIEmeter ranks the movies based on the searches done by the IMDB users and provides a good overview on which movies people are interested in. The two other movies, namely Sanctum and Gnomeo&Juliet, were also in the top 250. The DVD release dates for the movies were taken from a website called Movie Insider [26].

We used data on all the torrents uploaded to the Pirate Bay between the end of December, 2010 to the end of October, 2011. We used this data to check how fast new torrents of the feature films selected were uploaded to the Pirate Bay web site. We compared the release dates of the movies with the upload times of the torrents.

We used both the release dates of the movies in cinemas and the release dates of the DVD/Blu-ray versions of the movies and compared them to see how fast different versions of the movies showed up on the Pirate Bay.

Different versions of the movies are released at different times. For instance, usually the first releases are marked as CAM -releases. This means that the version of the movie has been recorded by a video camera, for example, in a movie theater. The version information can be found in the torrent name or the description page of the torrent. Below is a list of some of the different possible abbreviations used in torrent files to describe their quality.

CAM A torrent marked with a CAM means that the movie has been recorded in a movie theater with a digital video camera. This means that the camera may shake or move during the filming which affects the quality. Also, the sound is recorded with the camera's own microphone. The quality of CAM releases varies, but they usually have pretty poor video and sound quality. Sometimes background noises from the cinema can be heard. CAM versions of movies are usually the first ones to be released.

Telesync (TS) Telesync is like a CAM, but it uses a direct connection to an external sound source. The sound source can be, for example, the source used by the hearing-impaired. The video is often recorded with a professional video camera straight from the projector booth. Telesync releases are better quality than CAM releases, but they are still far from the quality of DVD or Blu-ray rips. Sometimes a CAM is mislabeled as a telesync. Telesync versions are released at the same time or right after CAM versions.

Telecine (TC) In a telecine release the movie has been copied digitally from the reel. Telecine is just a bit worse quality than DVD, because the source reel is not as good quality as the original source.

R5 R5 refers to DVD region 5. DVD region codes are designed to allow the film makers to distribute their movies in different regions at different times. DVD players have the regional-playback control (RPC) system which allows the player to play only discs encoded for that region. This way movies released in a region should not be viewable in another region. Region 5 consist of India, Africa, Russia and former USSR countries.

R5 releases are hurried out to decrease the level of piracy in region 5. R5 releases do not contain the same post-processing and clean up as DVDs so their quality is a

bit poorer. R5 copies are released in Russian by default so English audio has to be added from an external source. R5 releases are usually the next ones to show up on the Pirate Bay after telesync releases.

Pay-Per-View Rip (PPVRip) Pay-Per-View movies are available for buying in hotel rooms. New movies are often available before they are released on DVD or Blu-ray.

Screener A screener is an early DVD or BR version of the film sent to movie reviewer and executives for review purposes. There is usually a message of some sort on top of the picture stating that it is a promotional copy. The screener versions may contain scenes shown in black-and-white, but other than that the quality is close to that of the retail DVD.

DVD/Blu-ray Rip DVD or Blu-ray rips are the highest quality releases of the movies. They are ripped from the retail version of the film. These releases are the last ones to come. After a high quality DVD/BR -rip has been released there usually will not be any new releases. The "competition" on who releases the movies first is over.

There are more abbreviations used in the torrent names like, but these are the most common ones. In this research the torrents that come up are mostly telesync and DVD/BR-rip releases. Usually the first release of movie is a CAM or nowadays more often a telesync release. The first good quality release of a movie is generally a DVD rip and it is often the last one as well. However, there could be more releases with different resolutions uploaded.

5.2 The Publishing Speed of Movies on the Pirate Bay

Torrents of pretty much all of the most popular new movies are uploaded to the Pirate Bay web site. The speed at which the movies show up on the Pirate Bay web site varies, but the release schedule mostly follows the same pattern from movie to movie.

A significant portion of the uploaded torrents are fakes. Fake torrents are uploaded by anti-piracy agencies and malicious users. The Pirate Bay does a good job of removing the torrents whose description does not match the content. Users can report fake torrents in the forum or in the comments section. Fake torrents are identified by checking if the torrent still exists on the Pirate Bay web site.

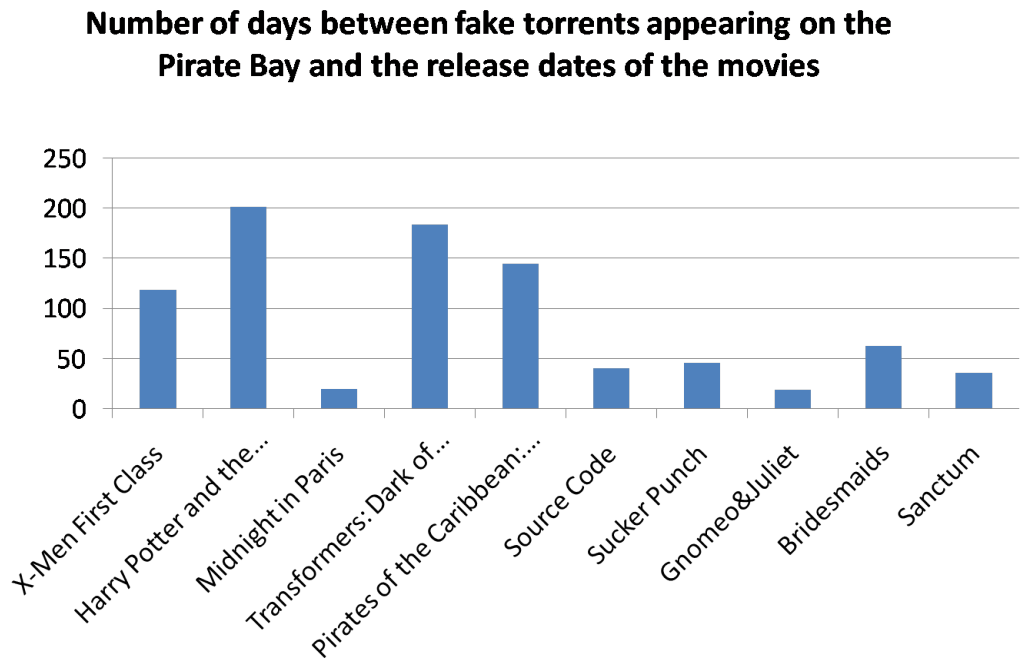

Figure 31: Graph shows how many days before fake torrents of the selected movies are uploaded to the Pirate Bay than the premiere dates of the movies are.

As can be seen in Figure 31, fake torrents are uploaded to the Pirate Bay as early as months before the premiere of the movie in question. These releases are often labeled as DVDRips, but they have been identified as fakes and removed form the Pirate Bay web site. In some cases, the same fake torrent is uploaded repeatedly as soon as it has been removed. Even though the Pirate Bay removes the fake torrents as soon as they are identified, many people have time to download the fake content.

The first legit torrent of a selected movie is uploaded around the time of the premiere of the movie, as is depicted in Figure 32. These releases are telesync releases, which means that they are recorded in a movie theater. For some of the movies the telesync versions appear on the Pirate Bay on the same day as the movie is first shown in cinemas. For most of the selected movies the interval is a just a few days, but sometimes even a few weeks.

The final and the best quality releases are DVD or Blu-ray rips. Figure 33 shows that the DVD rips are uploaded on the pirate bay usually a few weeks before the DVD is released. However, the DVD release dates used are for Region 1, meaning the United States and Canada. DVDs could be released in other regions earlier, so the uploaders can get the retail versions of the films from elsewhere.

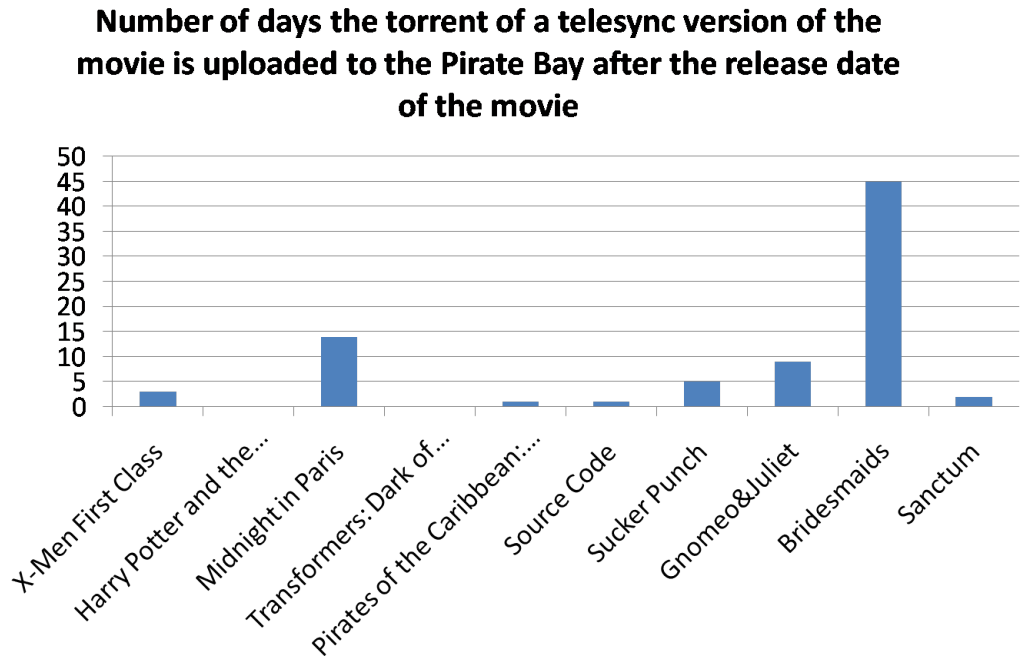

Figure 32: Graph shows how many days after the premiere date of the movie the torrent of telesync version for each of the selected movies are uploaded to the Pirate Bay web site.

There seems to be some competing on who uploads the torrent of a movie first. Three usernames, who have been identified as the biggest uploaders of torrents to the Pirate Bay, seem to be among the first ones in releasing the torrents of the selected ten movies. They have been tagged as VIP users, which means that they release a lot of good torrents which people are happy with. They are advertising their own private BitTorrent trackers on the description page of their releases. This seems to be paying off since they are releasing many torrents every day.

Most likely the same usernames are used by groups of people, because they upload so many torrents each day. In the ten movies selected, the three usernames are among the first ones to release the different versions of the movies. Most often the first telesync, R5 and DVD rip releases came from one of the big uploaders.

5.3 Evaluation

We could not find any research on the publishing speed of torrents of new movies to the Pirate Bay, so this could be the first one. We only selected ten movies for

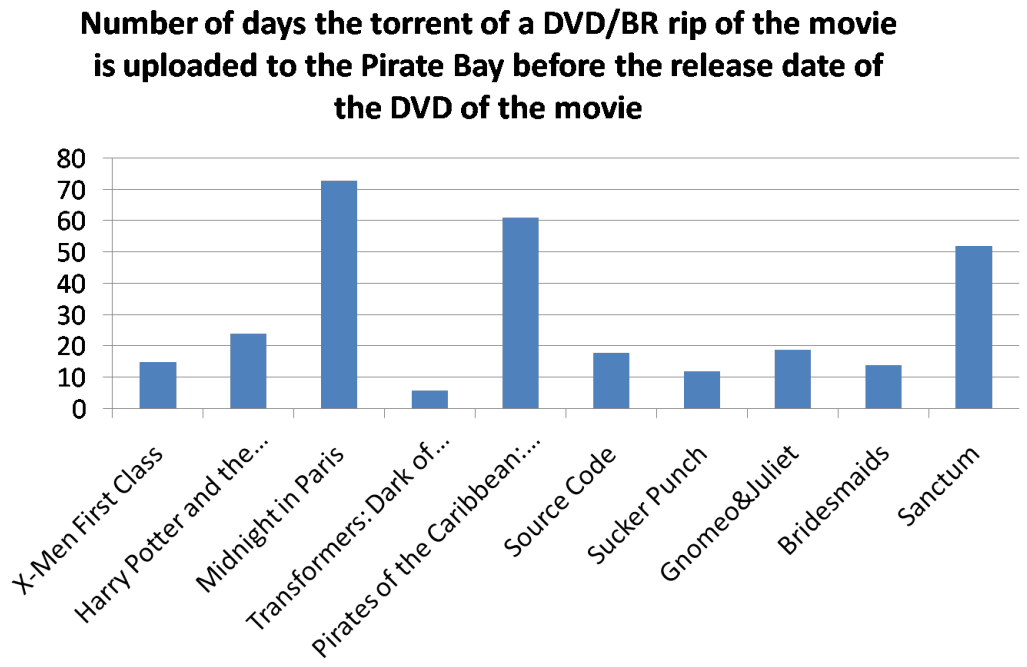

Figure 33: Graph shows how many days before the release date of the DVD/BR discs of the movies the torrents of DVD or Blu-ray versions of the selected movies are uploaded to the Pirate Bay.

this research so it might not be statistically valid. However, the results for all ten movies were very similar.

According to our investigation, it looks like there are three cases happening during the publishing of different torrents of a movie. First, even before the movie is released, the fake torrents of the movie start to appear on the Pirate Bay web site. Second, around the time of the premiere date of the movie, the telesync releases start to show up. Finally, the DVD rips emerge. The phases of the life cycle of the publishing of a movie on the Pirate Bay can be seen in table 14.

At first, the amount of fake torrents uploaded to the Pirate Bay, and the fact that they are uploaded so much before the movie is even released, may seem a bit odd. However, when you consider that people might not know the release dates of the new movies, it is not that startling. People may have heard the name of a movie before and see it on the Pirate Bay web site, so they excitedly download it without thinking it could be fake. In doing so, they could be installing malicious software to their computers. The film studios themselves could be uploading these fake torrents in an effort to advertise their movies.

Phase	Description	Days
1	The fake torrents start appearing.	200-10 days before release date
2	The premiere of the movie.	
3	Telesync versions appear	0-45 days after release date
4	DVD rips appear.	5-73 days before DVD release date
5	DVD release date.	

Table 14: The life cycle of the release of torrents for a movie.

The fact, that cam or telesync releases turn up on the Pirate Bay on the same day or just a few days after the premiere of the movie in cinemas, is not that surprising either. Good quality video cameras are fairly cheap to buy and it is not that hard to record a movie using a video camera in a dark movie theater. Also, the projector booth operators could be greased to let them record the movie straight from the booth using, for instance, a tripod to get better quality. This way it is also easy to record the audio from an external source using a FM transmitter intended for the hearing-impaired.

DVD or Blu-ray rips are very simple to make using programs designed for it. Copy protections can be circumvented. Hence, it is not surprising that DVD or Blu-ray rips are uploaded to the Pirate Bay as soon as the source is available.

Figure 34 [23] shows the amount of downloaded copies of movies by the methods of digitalization. The figure shows that the poor quality versions of the movies are not downloaded that much compared with the good quality DVD rip versions.

The same three usernames, TvTeam, sceneline and scenebalance, are among the first ones uploading different versions of the selected movies to the Pirate Bay. This implies that they want visibility to their uploaded torrents and they are obviously gaining something from this. They are advertising their own BitTorrent portals on the description pages of the torrents. An example of a description of one of scenebalance's torrents can be seen in Figure 35. Apparently they get people to sign up to their web sites and are making money from donations or membership fees.

In her research [22], Martikainen studied the short-term effects of file-sharing on DVD sales. It seems that the movies which are successful at the box office sell more in DVDs and are also downloaded a lot in BitTorrent. However, some movies which are selling a lot in DVDs have not been downloaded much in BitTorrent, and vice versa. She concludes, that file-sharing does not seem to reduce the sales of DVDs

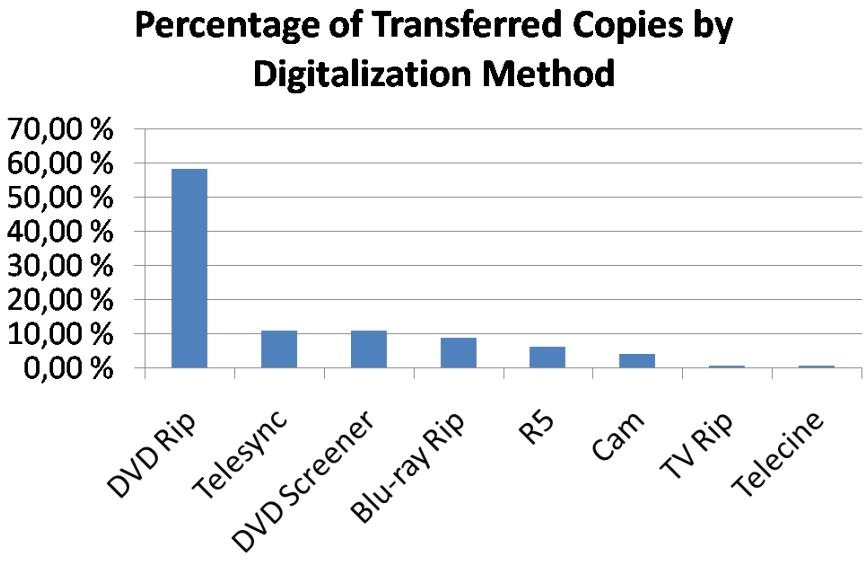

Figure 34: Percetages of copies transferred using BitTorrent by the digitalization method.

```
.

http://www.scenetime.com    -  Hottest private tracker on the net. !!!

Sign up will close shortly !!  Join now

        Seed back for speed


.....................................NFO:.................................................Movie
info :
GENRE.........: Action | Adventure | Drama
IMDB RATING...: 8.0/10 (83,836 votes)
LINK..........: http://www.imdb.com/title/tt1270798/

Release info :
RELEASE NAME..: X-Men.First.Class.2011.BRRip.XviD.RoSubbed-playXD
RELEASE SIZE..: 692 MB
RUNTIME.......: 2h 11min
VIDEO CODEC...: XviD
FRAMERATE.....: 23.976 fps
BITRATE.......: 691 Kbps
RESOLUTION....: 640 x 272 pixels
AUDIO.........: MP3 @ 128kbps
SUBTITLES.....: Romanian
ENCODER.......: iFeel
```

Figure 35: A screenshot of the description of a torrent uploaded by scenebalance.

of the movies in the short-term, but more research on the subject is needed.

The copyright holders of the movies seem to be concentrating on prosecuting users downloading the copyrighted content. The initial publishers seem to be out of reach for them, possibly hiding behind hosting providers or commercial ISPs, who will not give out give out their customer information. As long as this is the case, we do not see the end in publishing of the new movies on the Pirate Bay, as it seems to be profitable for the uploaders of the content.

6 Conclusion

In this paper, we looked into BitTorrent and how it is utilised on the Pirate Bay website. We used data on the torrents uploaded to the Pirate Bay during the time period of 25th of December, 2010 to 28th of October, 2011. We found out that a small percentage of users are responsible for a large portion of the uploaded torrents. These usernames are uploading so many torrents per day, some as much as 70 torrents per day, that there must be groups of people behind the usernames. The top publishers are often advertising a website on the description pages of the torrents. This must be paying off for them, since they have continued to upload so many torrents for such a long time.

We examined the publishing behaviour of the top publishers and it became obvious that the people behind the usernames prepare the torrents beforehand and then upload them in chunks. It could also be so that other people are creating the torrents while one person is adding them to the website. For some users, like exmnova, it seemed quite clear that they are using a script to upload the torrents, since their rate of uploading was so fast.

It became clear to us that torrents are constantly uploaded to the Pirate Bay website. We noticed that the rate of uploading of torrents to the Pirate Bay website is so fast, that from the total of 651 712 uploaded torrents, about 600 000 consecutive uploads, or about 92% of the uploads, have happened within 100 seconds or less from each other. Thus, any downtime in the Pirate Bay servers will affect the upload rate significantly.

Looking at the file size distributions of the torrents we discovered that most of the files, about 85% of the files, are less than 1.5 GB in size. This number can be expected to go up as software, audio and video files will grow in size, when more

detail is added.

Finally, we studied how fast torrents of new movies appeared on the Pirate Bay compared with the release dates of the movies. We found out that all the releases followed the same pattern. First fake versions of the movie appeared. Then just a few days after the premiere of the movie, the telesync versions showed up. Finally the DVD versions, which were often the final releases of the movies, appeared even before the official release dates of the DVDs in most regions.

There seems to be some competition on who uploads the torrents of the new movies first. The top publishers seem to be the ones uploading the popular torrents to gain as much exposure as possible.

If one wants to elaborate on the whole BitTorrent ecosystem more thoroughly, more data, especially on the downloaders connected to the swarm, is needed. We only got data on the torrents uploaded to the Pirate Bay portal, which set some restrictions on what kind of information we could gather.

BitTorrent is quite an ingenious system in the distribution of files. With the enhancements made to the protocol we do not see it going anywhere. We provided information on the content being uploaded, the users uploading the content and how the uploads are made to the Pirate Bay website. This information should be useful for researchers and anyone wanting to create their own BitTorrent portal.

References

1 Alexa the Web Information Company, http://www.alexa.com/, [7.6.2012].

2 The Base16, Base32, and Base64 Data Encodings, http://www.ietf.org/rfc/rfc3548.txt, [6.6.2012].

3 A. R. Bharambe, C. Herley, V. N. Padmanabhan, Some Observations on BitTorrent Performance, Proc. ACM SIGMETRIC, 2005

4 The BitTorrent Protocol Specification, http://bittorrent.org/beps/bep_0003.html, [30.5.2012].

5 B. Cohen. Incentives Build Robustness in BitTorrent. May 22, 2003. http://www.bittorrent.org/bittorrentecon.pdf.

6 Compete, http://www.compete.com, [7.6.2012].

7 T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms (2nd edition). MIT Press and McGraw-Hill, p. 221-252, 2001.

8 R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie. Is Content Publishing in BitTorrent Altruistic or Profit-Driven? CoNEXT 2010.

9 DHT Protocol, http://www.bittorrent.org/beps/bep_0005.html, [5.6.2012].

10 eMule-Project.net - Official eMule Homepage, http://www.emule-project.net [7.8.2012].

11 Extension for Peers to Send Metadata Files, http://bittorrent.org/beps/bep_0009.html, [5.6.2012].

12 The Gnutella Protocol Specification v0.4, http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf, [6.8.2012].

13 O. Heckmann, A. Bock. The eDonkey 2000 Protocol. KOM Technical Report 08/2002, Version 0.8, December 2002.

14 Internet Study 2008/2009. http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf, [7.12.2011].

15 The Internet Movie Database, http://www.imdb.com/, [15.5.2012].

16 M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A. Al Hamra, L. Garces-Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. PAM, April, 2004.

17 JFreeChart. http://www.jfree.org/jfreechart/ [3.9.2012].

18 Y. Kulbak, D. Bickson. The eMule Protocol Specification. http://www.cs.huji.ac.il/labs/danss/p2p/resources/emule.pdf January 17, 2005.

19 S. Le Blond, A. Legout, F Lefessant, W. Dabbous, M. Ali Kafaar. Spying the World From Your Laptop - Identifying and Profiling Content Providers and Big Downloaders in BitTorrent, Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'10), April 27, 2010.

20 A. Legout, G. Urvoy-Keller, P. Michiardi. Rarest First and Choke Algorithms Are Enough, Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Pages 203 - 216, 2006.

21 J. Liang, R. Kumar, K. W. Ross. Understanding KaZaA. http://cis.poly.edu/ ross/papers/UnderstandingKaZaA.pdf, 2004.

22 E. Martikainen. Does File-Sharing Reduce DVD Sales? January 17, 2011. Available at SSRN: http://ssrn.com/abstract=1742443 or http://dx.doi.org/10.2139/ssrn.1742443.

23 A. M. Mateus, J. M. Peha. Quantifying Global Transfers of Copyrighted Content Using BitTorrent. TPRC 2011. Available at SSRN: http://ssrn.com/abstract=1985737.

24 P. Maymounkov, D. Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, Proceedings of the 1st International Workshop on Peer-to-Peer Systems, 2002.

25 The MD4 Message-Digest Algorithm. http://tools.ietf.org/html/rfc1320 [8.8.2012].

26 Movie Insider, www.movieinsider.com, [15.5.2012].

27 The Pirate Bay, http://thepiratebay.se/, [25.5.2012].

28 J. A. Pouwelse, P. Garbacki, D. H. J. Epema, H. J. Sips. A Measurement Study of the BitTorrent Peer-to-Peer File-Sharing System. Technical Report PDS-2004-003, Delft University of Technology, The Netherlands, April 2004.

29 RFC 2141 - URN Syntax, http://tools.ietf.org/html/rfc2141, [6.6.2012].

30 S. Saroiu, P. K. Gummadi, S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001.

31 R. Schollmeier. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, p2p, pp.0101, First International Conference on Peer-to-Peer Computing (P2P'01), 2001.

32 TorrentFreak, http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/, [23.3.2012].

33 TorrentFreak, http://torrentfreak.com/top-10-most-popular-torrent-sites-of-2012-120107/, [25.5.2012].

34 C. Zhang, P Dhungle, D. Wu, K. W. Ross. Unraveling the BitTorrent Ecosystem. IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 7, July 2011.

A SQL queries

Following is a list of the SQL queries used to fetch data from the database.

Major publishers:

```
SELECT * FROM (SELECT publisher, COUNT(*) count FROM publish GROUP BY publisher WITH ROLLUP) t ORDER BY count DESC
```

Content published:

```
SELECT type, COUNT(*) FROM publish GROUP BY type
```

Torrents published by a publisher. Username is the publishers username.

```
SELECT * FROM publish where publisher = '[username]' order by uploadtime
```

Uploads per day:

```
SELECT COUNT(*) as updatesPerDay, DATE(uploadtime) AS date FROM publish GROUP BY date ORDER BY date
```

Sizes of files. Type is one of the types. For all sizes remove 'where type = [type]'.

```
SELECT size FROM publish where type = '[type]'
```

Uploads per day. Type was replaced by publisher to get the publisher's uploads.

```
SELECT COUNT(*) as updatesPerDay, DATE(uploadtime) AS date FROM publish where type =' [type]' GROUP BY date ORDER BY date
```

Uploads per weekday. 'publisher = [publisher]' was removed to get all of the uploads.

```
SELECT weekday(uploadtime) as DayOfWeek, count(weekday(uploadtime)) as updatesPerDay FROM publish where publisher = '[publisher]' group by DayOfWeek
```

Uploadtimes used to count the time intervals between uploadtimes:

```
SELECT uploadtime FROM publish where publisher = '[publisher]' order by uploadtime desc
```

Uploads per hour of the day:

```
SELECT extract(hour from uploadtime) as hour, count(extract(hour from uploadtime)) as updatesPerHour FROM publish where type = '[type]' group by hour
```

Top publishers:

```
SELECT * FROM (SELECT publisher, COUNT(*) count FROM publish GROUP BY publisher WITH ROLLUP) t ORDER BY count DESC
```

Movies:

```
SELECT * FROM publish WHERE type = '[type]' AND subtype LIKE '
```